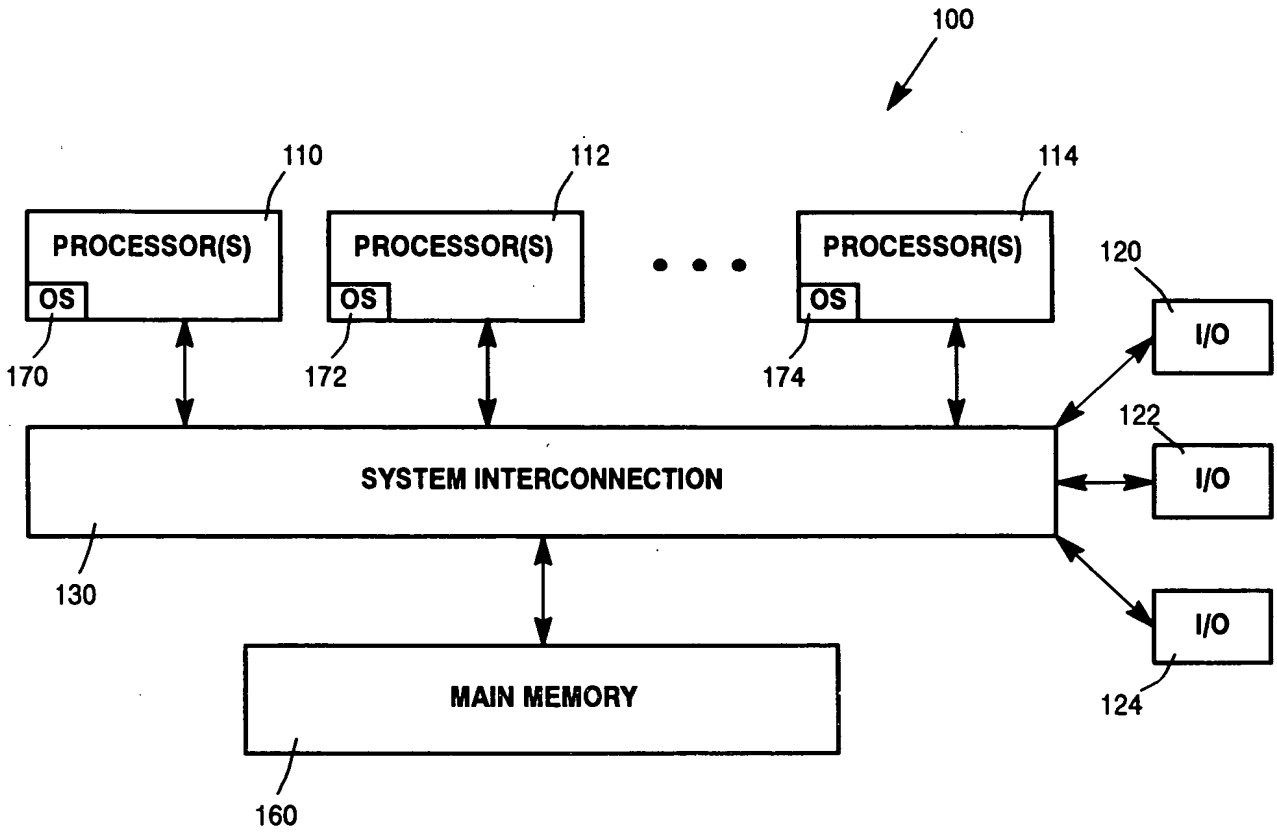
100
110
112
114
PROCESSOR(S)
OS
PROCESSOR(S)
OS
PROCESSOR(S)
OS
120
I/O
170
172
174
122
SYSTEM INTERCONNECTION
I/O
130
I/O
MAIN MEMORY
124
160

Figure 1

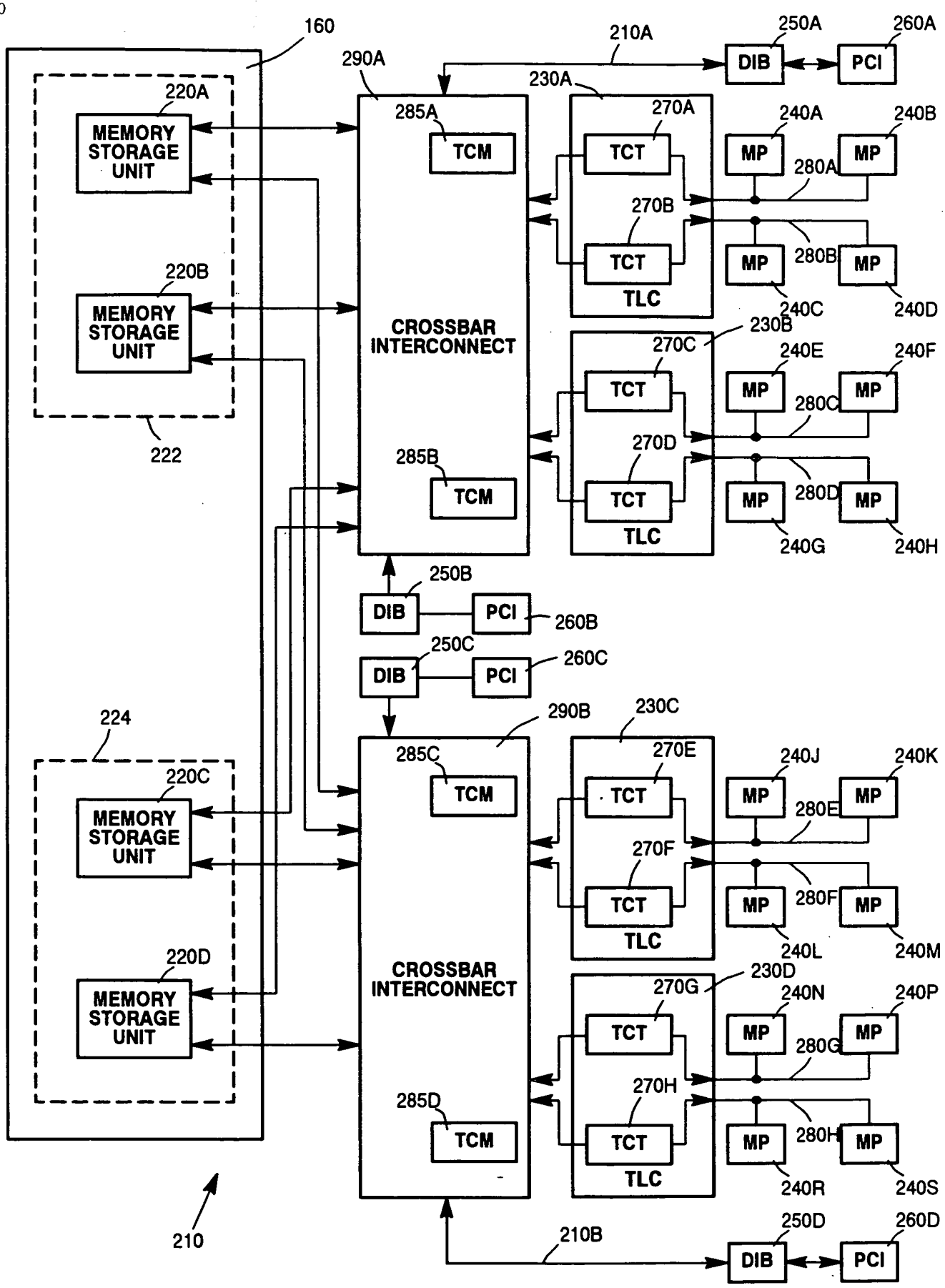
160
210A
250A
260A
DIB
PCI
290A
230A
220A
MEMORY STORAGE UNIT
285A
TCM
270A
TCT
240A
240B
MP
280A
270B
TCT
TLC
280B
240C
240D
220B
MEMORY STORAGE UNIT
CROSSBAR INTERCONNECT
230B
270C
240E
240F
280C
270D
222
285B
TCM
280D
240G
240H
250B
DIB
PCI
260B
250C
DIB
PCI
260C
224
290B
230C
220C
MEMORY STORAGE UNIT
285C
TCM
270E
240J
240K
280E
270F
280F
240L
240M
220D
MEMORY STORAGE UNIT
CROSSBAR INTERCONNECT
230D
270G
240N
240P
280G
270H
285D
TCM
280H
240R
240S
210
210B
250D
260D
DIB
PCI

Figure 2

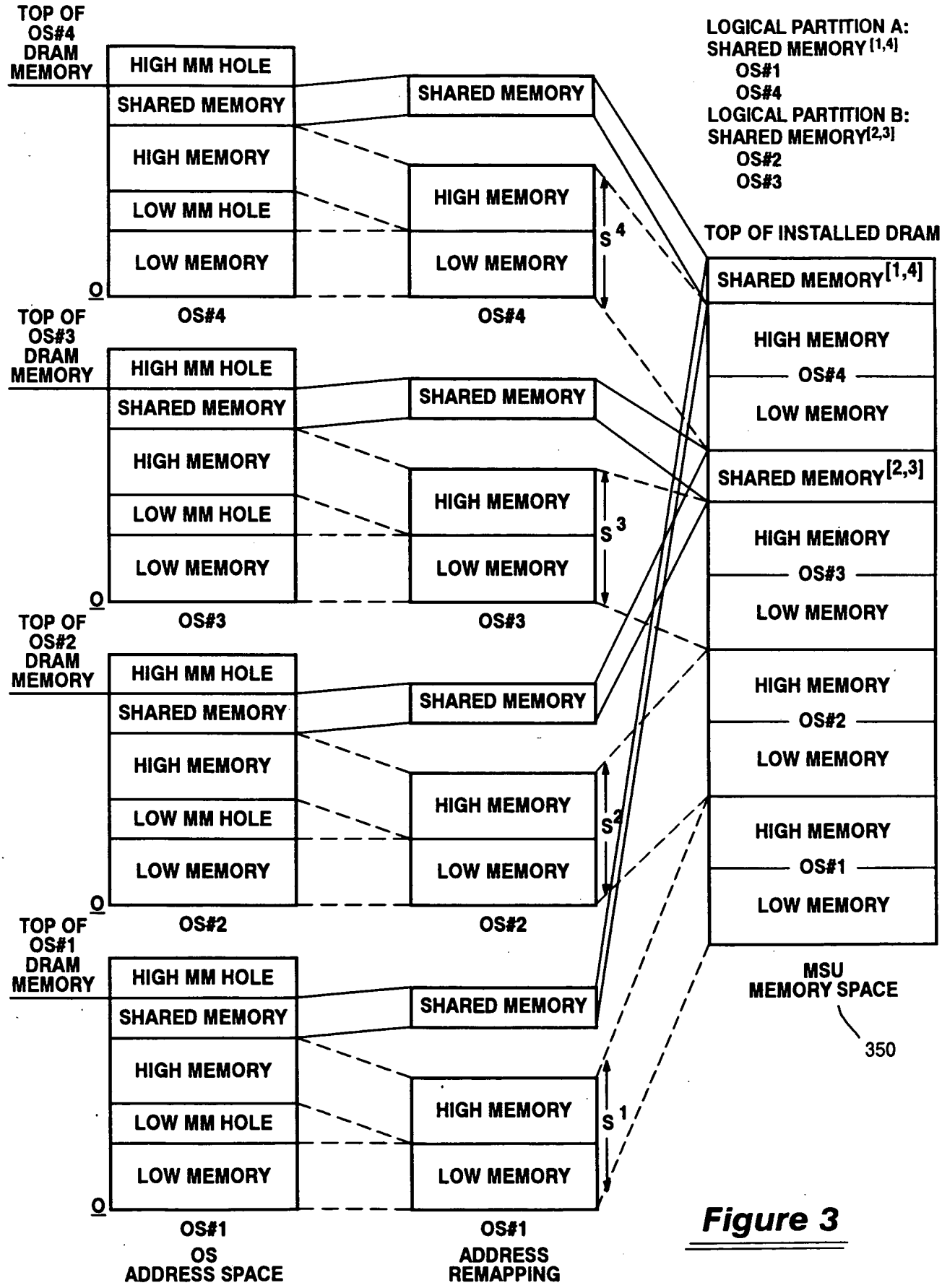
TOP OF OS#4 DRAM MEMORY
HIGH MM HOLE
SHARED MEMORY
HIGH MEMORY
LOW MM HOLE
LOW MEMORY
0
OS#4
OS#4
TOP OF OS#3 DRAM MEMORY
OS#3
OS#3
TOP OF OS#2 DRAM MEMORY
OS#2
OS#2
TOP OF OS#1 DRAM MEMORY
OS#1
OS#1
OS ADDRESS SPACE
ADDRESS REMAPPING
S4
S3
S2
S1
LOGICAL PARTITION A:
SHARED MEMORY[1,4]
OS#1
OS#4
LOGICAL PARTITION B:
SHARED MEMORY[2,3]
OS#2
OS#3
TOP OF INSTALLED DRAM
SHARED MEMORY[1,4]
HIGH MEMORY
OS#4
LOW MEMORY
SHARED MEMORY[2,3]
HIGH MEMORY
OS#3
LOW MEMORY
HIGH MEMORY
OS#2
LOW MEMORY
HIGH MEMORY
OS#1
LOW MEMORY
MSU MEMORY SPACE
350

Figure 3

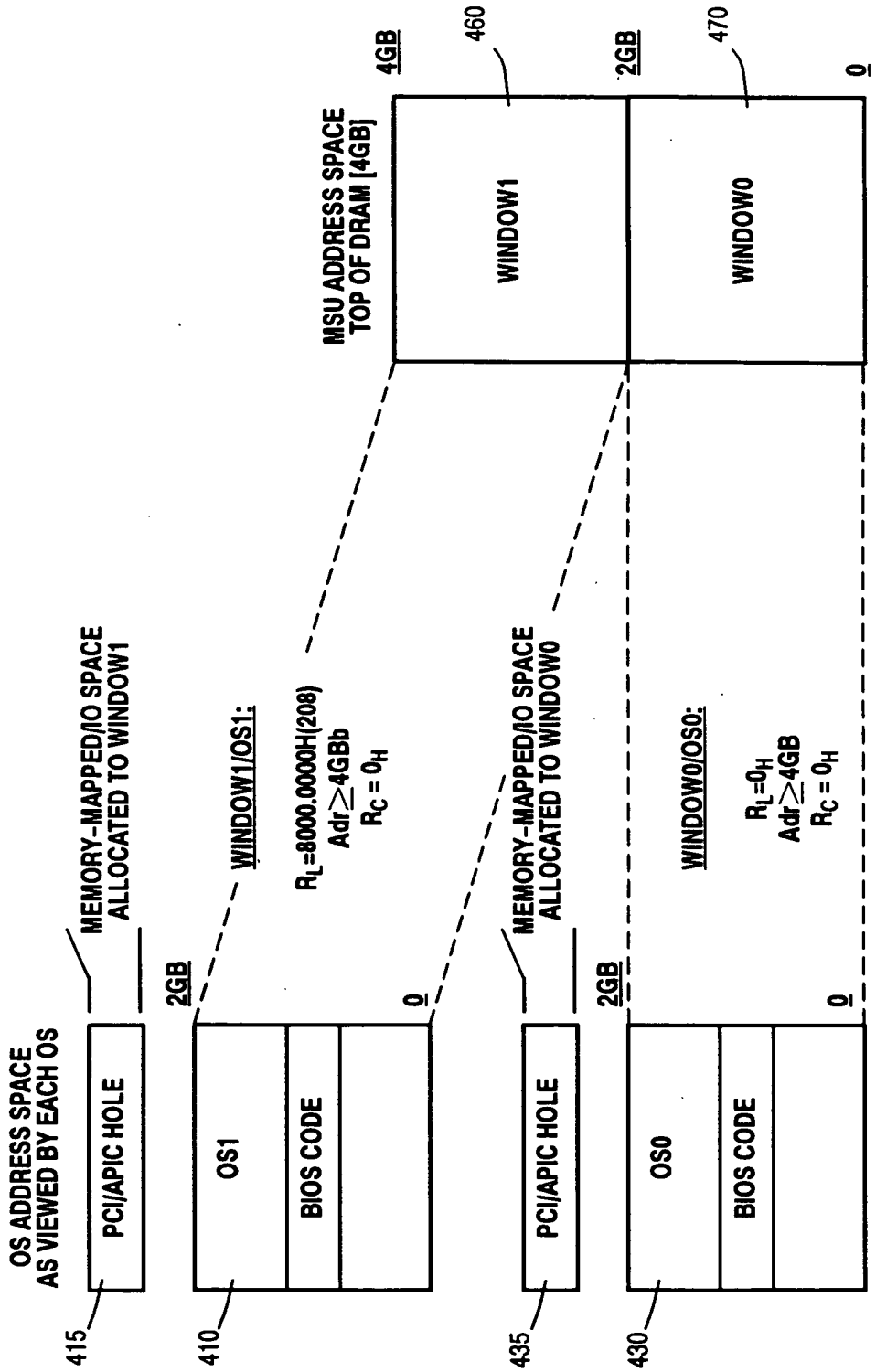
OS ADDRESS SPACE
AS VIEWED BY EACH OS
MEMORY-MAPPED/IO SPACE
ALLOCATED TO WINDOW1
PCI/APIC HOLE
415
2GB
0
WINDOW1/OS1:
R_L=8000.0000H(208)
Adr≥4GBb
$R_C = 0_H$
OS1
BIOS CODE
410
MEMORY-MAPPED/IO SPACE
ALLOCATED TO WINDOW0
PCI/APIC HOLE
435
WINDOW0/OS0:
$R_L = 0_H$
Adr≥4GB
$R_C = 0_H$
OS0
BIOS CODE
430
MSU ADDRESS SPACE
TOP OF DRAM [4GB]
4GB
460
WINDOW1
2GB
470
WINDOW0
0

Figure 4

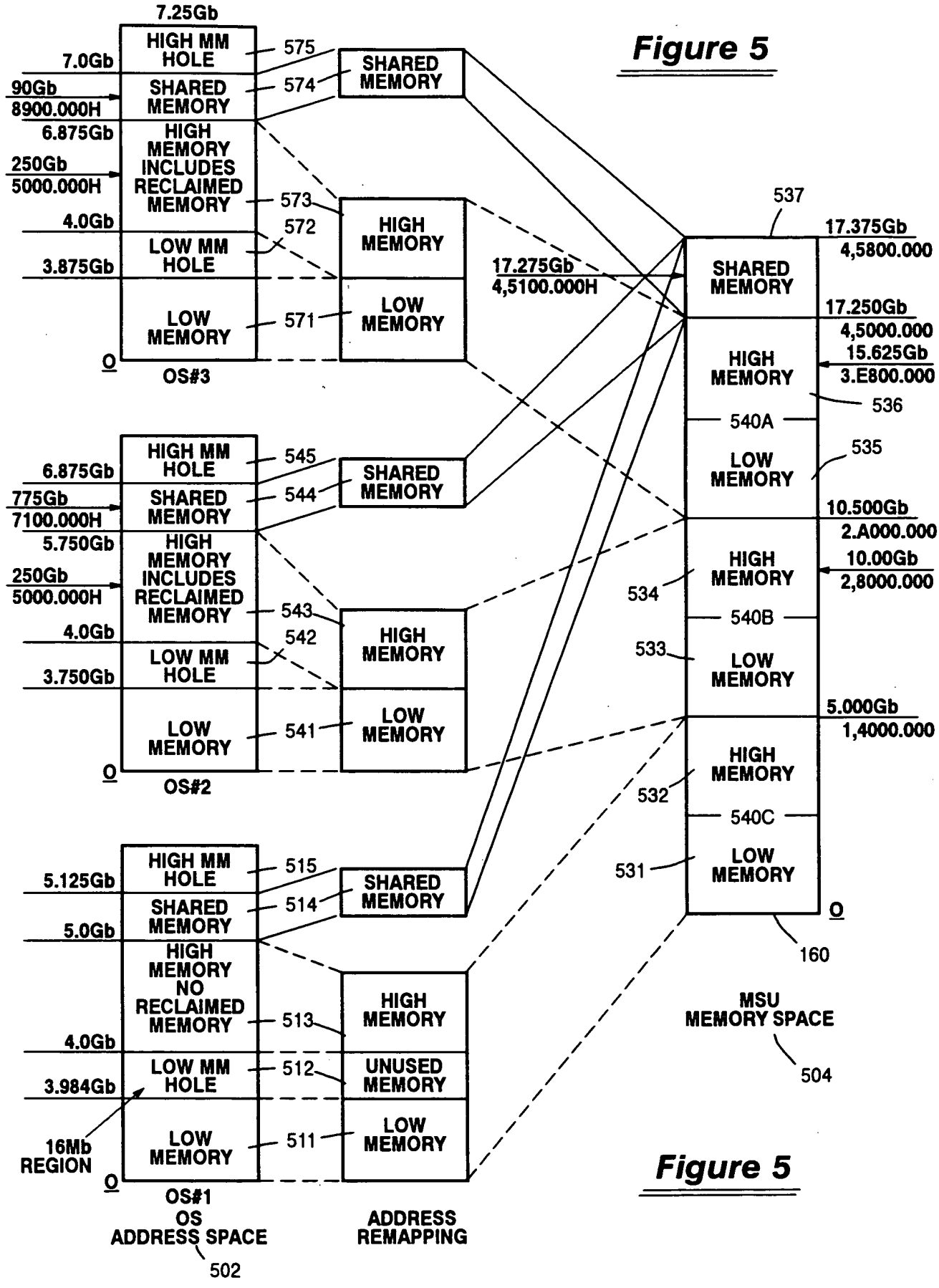
Figure 5
7.25Gb
HIGH MM HOLE
7.0Gb
SHARED MEMORY
90Gb
8900.000H
6.875Gb
HIGH MEMORY INCLUDES RECLAIMED MEMORY
250Gb
5000.000H
4.0Gb
LOW MM HOLE
3.875Gb
LOW MEMORY
0
OS#3
575
574
573
572
571
SHARED MEMORY
HIGH MEMORY
LOW MEMORY
17.275Gb
4,5100.000H
537
17.375Gb
4,5800.000
SHARED MEMORY
17.250Gb
4,5000.000
HIGH MEMORY
15.625Gb
3.E800.000
536
540A
LOW MEMORY
535
10.500Gb
2.A000.000
HIGH MEMORY
10.00Gb
2,8000.000
534
540B
533
LOW MEMORY
5.000Gb
1,4000.000
HIGH MEMORY
532
540C
LOW MEMORY
531
0
160
MSU MEMORY SPACE
504
HIGH MM HOLE
6.875Gb
SHARED MEMORY
775Gb
7100.000H
5.750Gb
HIGH MEMORY INCLUDES RECLAIMED MEMORY
250Gb
5000.000H
4.0Gb
LOW MM HOLE
3.750Gb
LOW MEMORY
0
OS#2
545
544
543
542
541
SHARED MEMORY
HIGH MEMORY
LOW MEMORY
HIGH MM HOLE
5.125Gb
SHARED MEMORY
5.0Gb
HIGH MEMORY NO RECLAIMED MEMORY
4.0Gb
LOW MM HOLE
3.984Gb
16Mb REGION
LOW MEMORY
0
OS#1
OS ADDRESS SPACE
502
515
514
513
512
511
SHARED MEMORY
HIGH MEMORY
UNUSED MEMORY
LOW MEMORY
ADDRESS REMAPPING
Figure 5

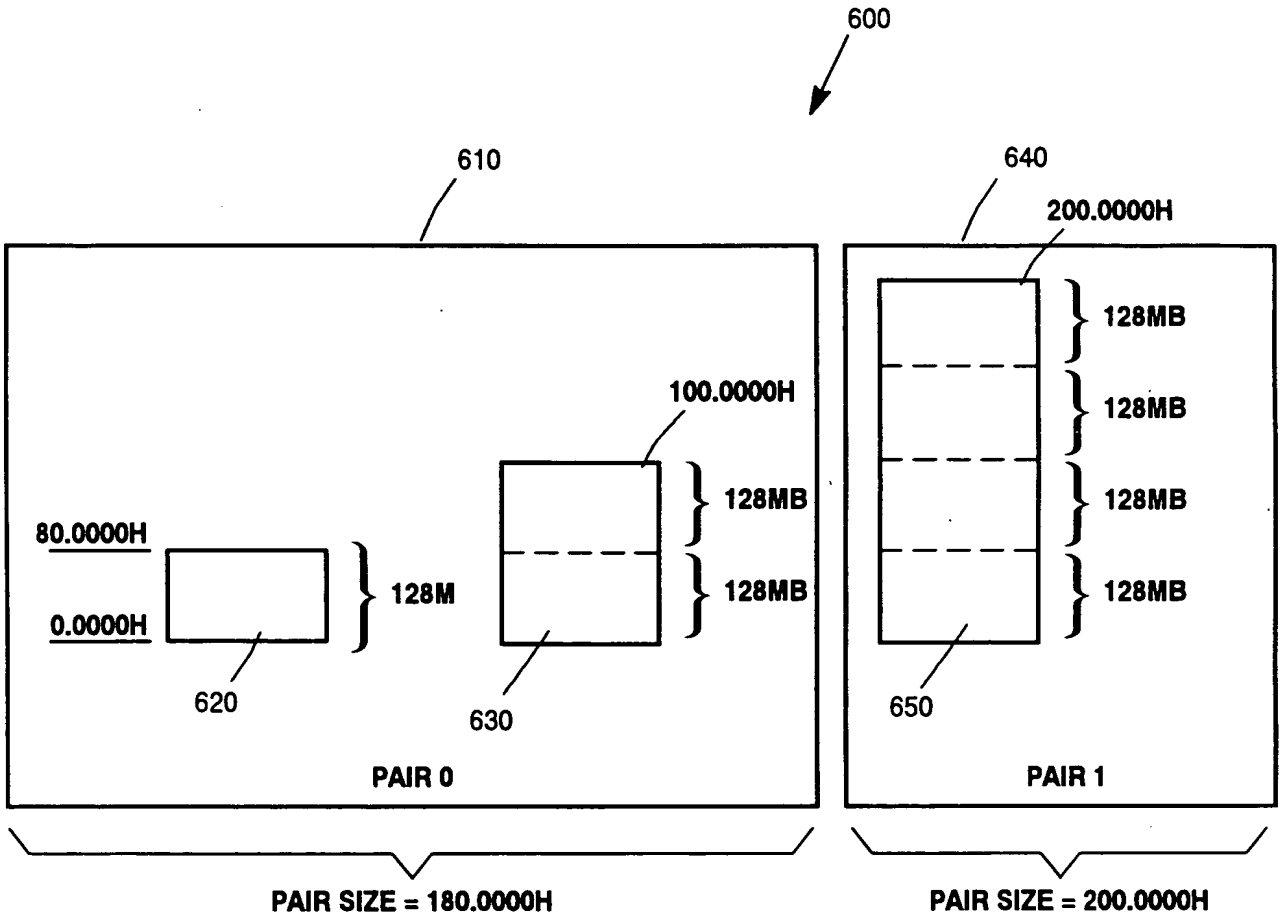
600
610
640
200.0000H
128MB
128MB
128MB
128MB
100.0000H
128MB
128MB
80.0000H
0.0000H
128M
620
630
650
PAIR 0
PAIR 1
PAIR SIZE = 180.0000H
PAIR SIZE = 200.0000H

Figure 6

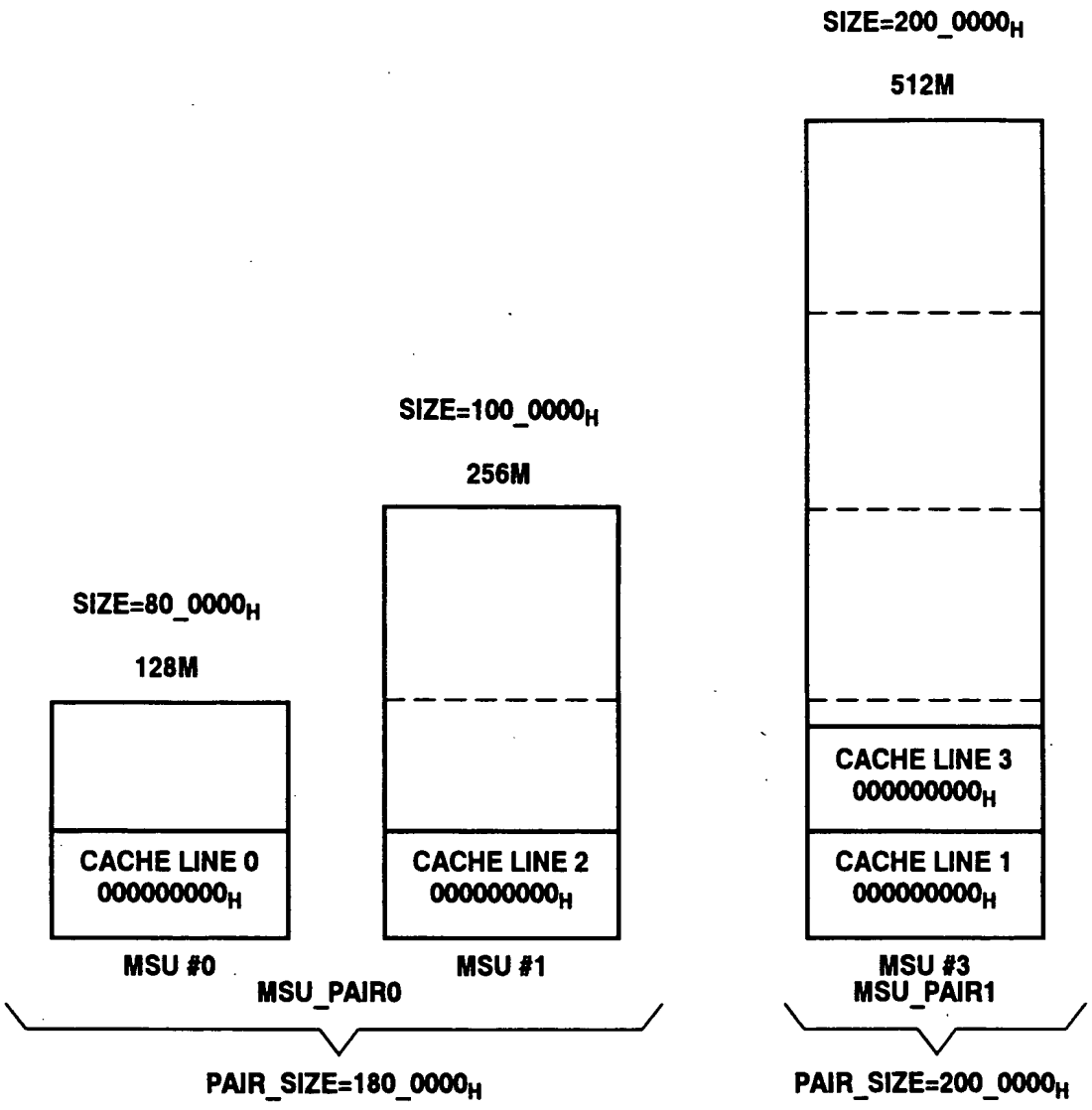
SIZE=200_0000H
512M
SIZE=100_0000H
256M
SIZE=80_0000H
128M
CACHE LINE 3
000000000H
CACHE LINE 1
000000000H
CACHE LINE 2
000000000H
CACHE LINE 0
000000000H
MSU #0
MSU #1
MSU #3
MSU_PAIR0
MSU_PAIR1
PAIR_SIZE=180_0000H
PAIR_SIZE=200_0000H

Figure 7

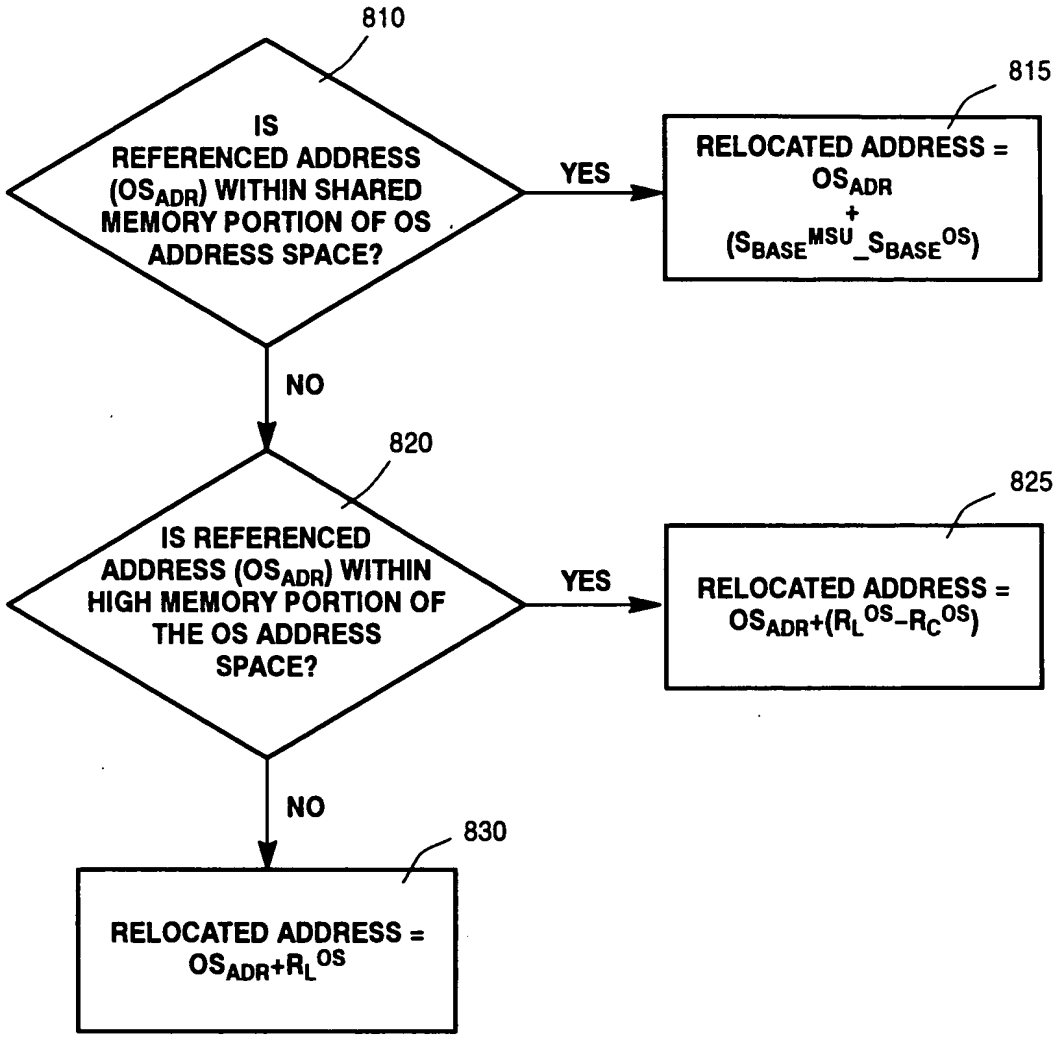
810
IS REFERENCED ADDRESS (OS_{ADR}) WITHIN SHARED MEMORY PORTION OF OS ADDRESS SPACE?
YES
815
RELOCATED ADDRESS = OS_{ADR} + (S_{BASE}^{MSU} _ S_{BASE}^{OS})
NO
820
IS REFERENCED ADDRESS (OS_{ADR}) WITHIN HIGH MEMORY PORTION OF THE OS ADDRESS SPACE?
YES
825
RELOCATED ADDRESS = OS_{ADR}+(R_L^{OS}-R_C^{OS})
NO
830
RELOCATED ADDRESS = OS_{ADR}+R_L^{OS}

Figure 8

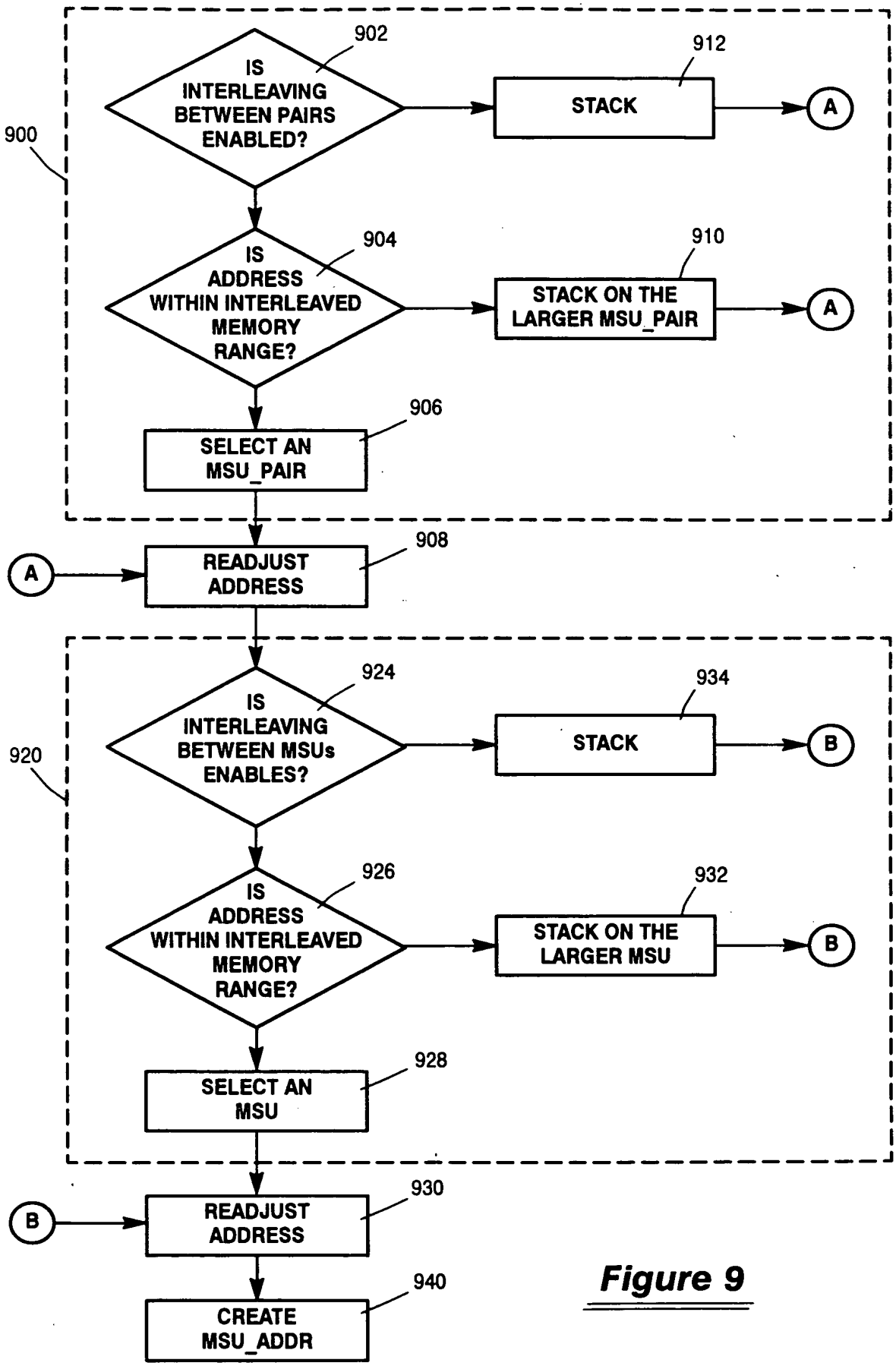
900
902
IS INTERLEAVING BETWEEN PAIRS ENABLED?
912
STACK
A
904
IS ADDRESS WITHIN INTERLEAVED MEMORY RANGE?
910
STACK ON THE LARGER MSU_PAIR
A
906
SELECT AN MSU_PAIR
A
908
READJUST ADDRESS
920
924
IS INTERLEAVING BETWEEN MSUs ENABLES?
934
STACK
B
926
IS ADDRESS WITHIN INTERLEAVED MEMORY RANGE?
932
STACK ON THE LARGER MSU
B
928
SELECT AN MSU
B
930
READJUST ADDRESS
940
CREATE MSU_ADDR

Figure 9

USYS-0094 "System and Method for Emulating Network Communications between Partitions of a Computer System"
S.B.Samuels-WoodcockWashburn et al.
215-568-3100

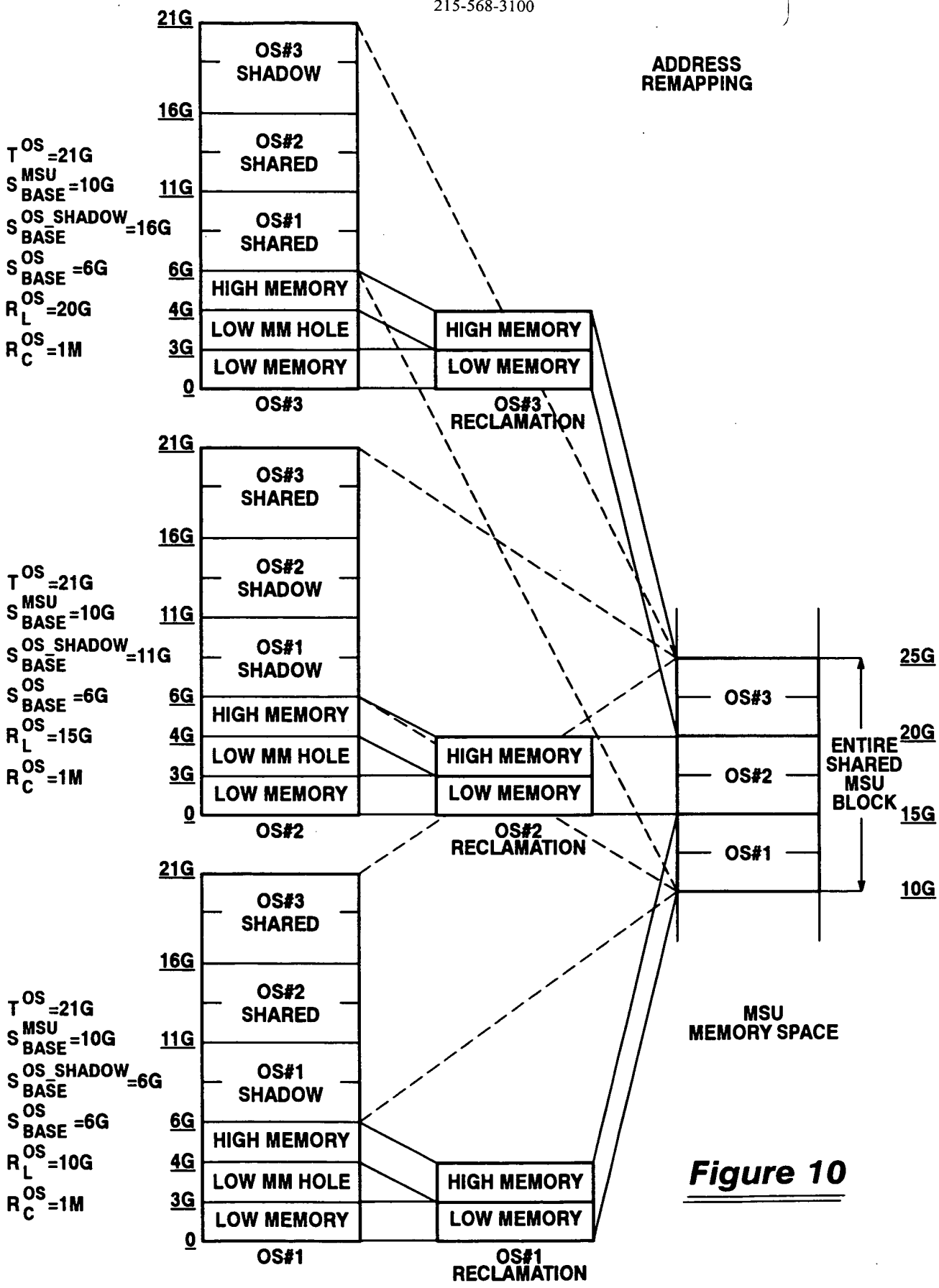

Figure 10

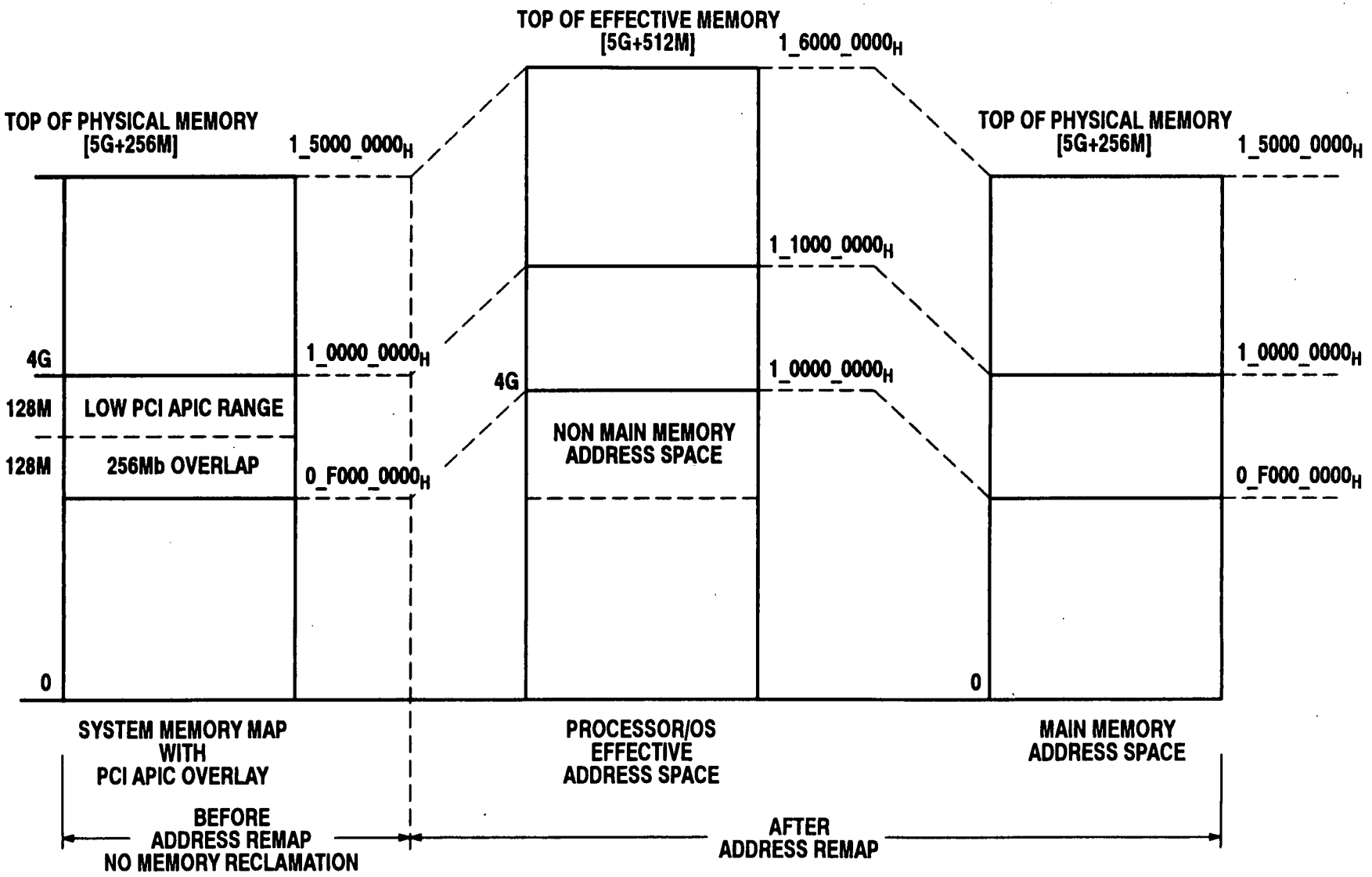
TOP OF EFFECTIVE MEMORY
[5G+512M]
1_6000_0000H
TOP OF PHYSICAL MEMORY
[5G+256M]
1_5000_0000H
TOP OF PHYSICAL MEMORY
[5G+256M]
1_5000_0000H
1_1000_0000H
4G
1_0000_0000H
4G
1_0000_0000H
1_0000_0000H
128M
LOW PCI APIC RANGE
128M
256Mb OVERLAP
0_F000_0000H
NON MAIN MEMORY
ADDRESS SPACE
0_F000_0000H
0
0
SYSTEM MEMORY MAP
WITH
PCI APIC OVERLAY
PROCESSOR/OS
EFFECTIVE
ADDRESS SPACE
MAIN MEMORY
ADDRESS SPACE
BEFORE
ADDRESS REMAP
NO MEMORY RECLAMATION
AFTER
ADDRESS REMAP

Figure 11

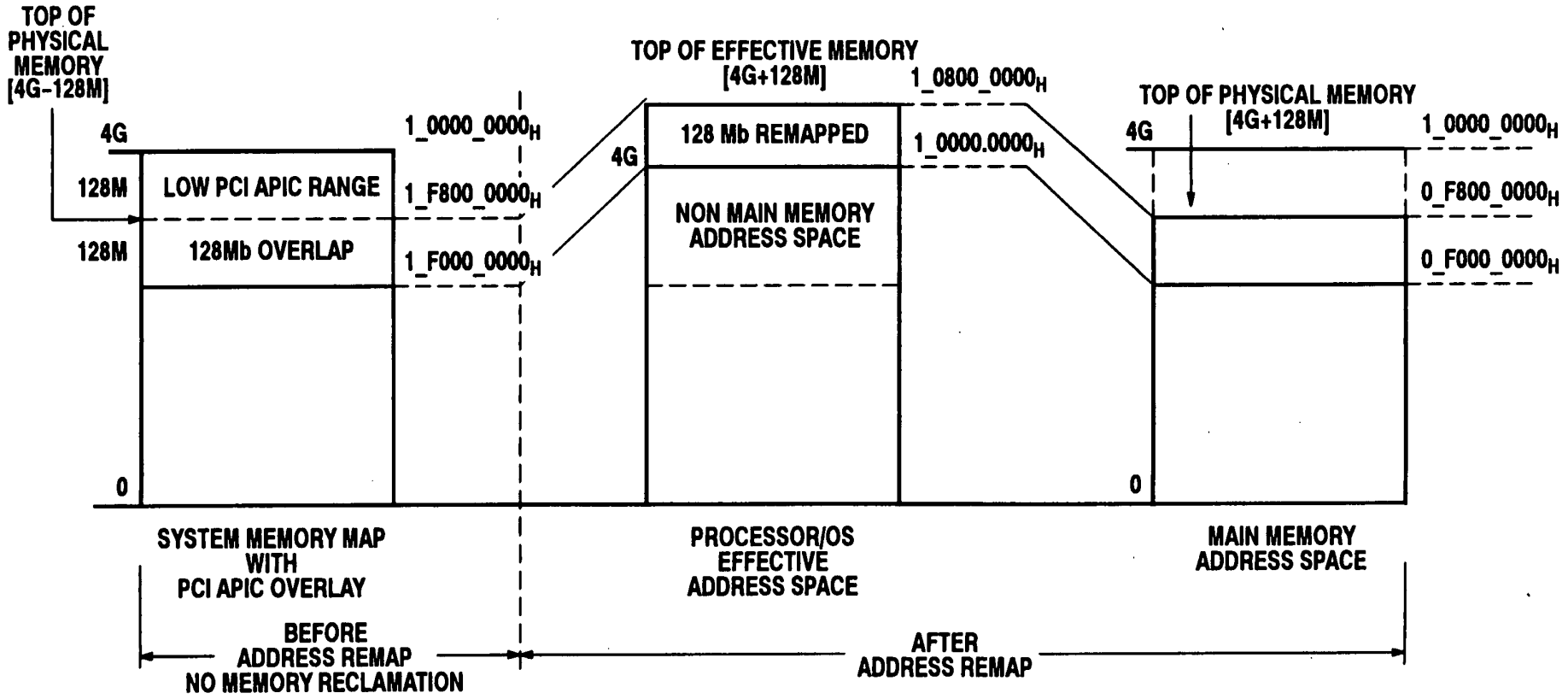
TOP OF PHYSICAL MEMORY [4G-128M]
4G
128M
LOW PCI APIC RANGE
128M
128Mb OVERLAP
0
1_0000_0000H
1_F800_0000H
1_F000_0000H
SYSTEM MEMORY MAP WITH PCI APIC OVERLAY
TOP OF EFFECTIVE MEMORY [4G+128M]
1_0800_0000H
4G
128 Mb REMAPPED
NON MAIN MEMORY ADDRESS SPACE
1_0000.0000H
PROCESSOR/OS EFFECTIVE ADDRESS SPACE
TOP OF PHYSICAL MEMORY [4G+128M]
4G
1_0000_0000H
0_F800_0000H
0_F000_0000H
0
MAIN MEMORY ADDRESS SPACE
BEFORE ADDRESS REMAP NO MEMORY RECLAMATION
AFTER ADDRESS REMAP

Figure 12

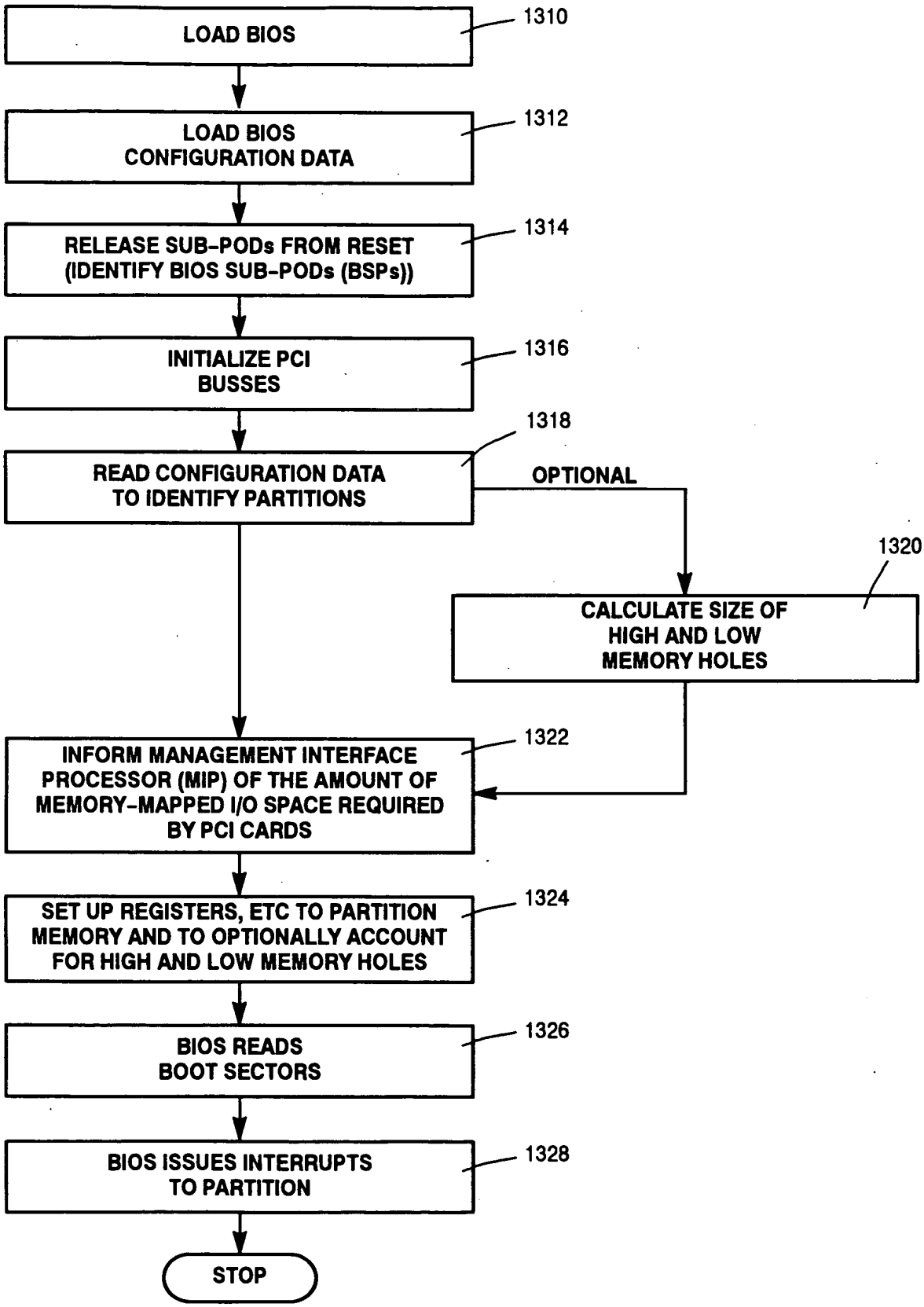
LOAD BIOS
1310
LOAD BIOS CONFIGURATION DATA
1312
RELEASE SUB-PODs FROM RESET (IDENTIFY BIOS SUB-PODs (BSPs))
1314
INITIALIZE PCI BUSSES
1316
READ CONFIGURATION DATA TO IDENTIFY PARTITIONS
1318
OPTIONAL
CALCULATE SIZE OF HIGH AND LOW MEMORY HOLES
1320
INFORM MANAGEMENT INTERFACE PROCESSOR (MIP) OF THE AMOUNT OF MEMORY-MAPPED I/O SPACE REQUIRED BY PCI CARDS
1322
SET UP REGISTERS, ETC TO PARTITION MEMORY AND TO OPTIONALLY ACCOUNT FOR HIGH AND LOW MEMORY HOLES
1324
BIOS READS BOOT SECTORS
1326
BIOS ISSUES INTERRUPTS TO PARTITION
1328
STOP

Figure 13

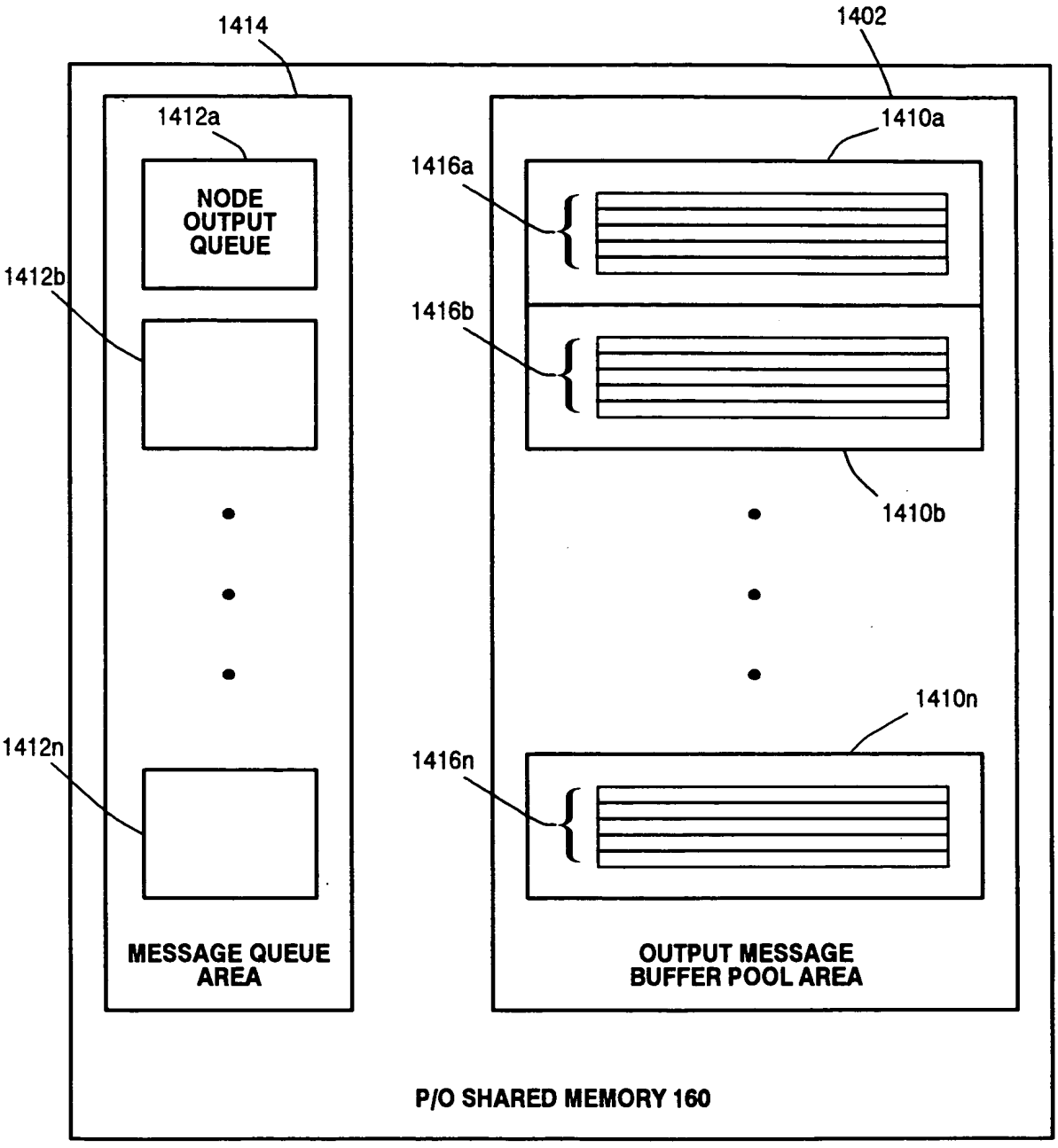
1414
1402
1412a
1410a
1416a
NODE
OUTPUT
QUEUE
1412b
1416b
1410b
1410n
1412n
1416n
MESSAGE QUEUE
AREA
OUTPUT MESSAGE
BUFFER POOL AREA
P/O SHARED MEMORY 160

Figure 14

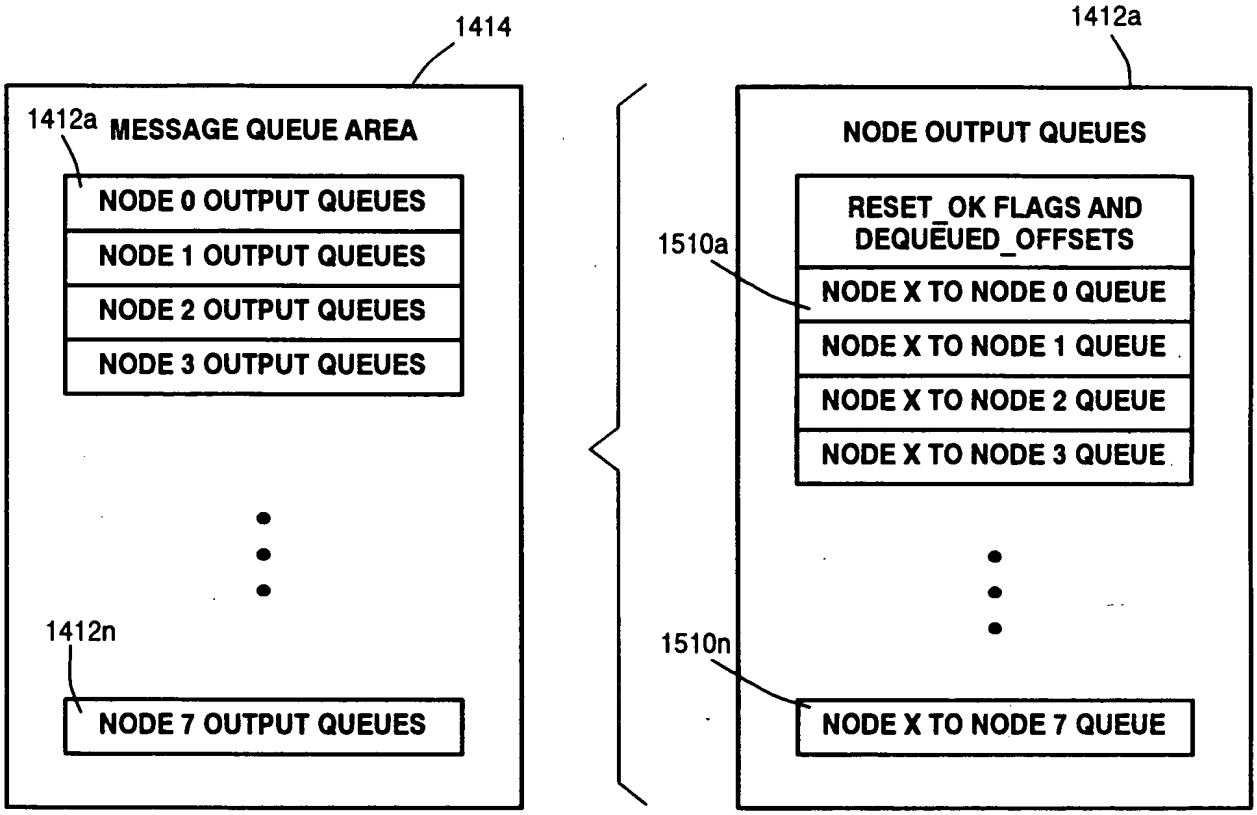
1414
1412a
MESSAGE QUEUE AREA
NODE 0 OUTPUT QUEUES
NODE 1 OUTPUT QUEUES
NODE 2 OUTPUT QUEUES
NODE 3 OUTPUT QUEUES
1412n
NODE 7 OUTPUT QUEUES
1412a
NODE OUTPUT QUEUES
RESET_OK FLAGS AND DEQUEUED_OFFSETS
1510a
NODE X TO NODE 0 QUEUE
NODE X TO NODE 1 QUEUE
NODE X TO NODE 2 QUEUE
NODE X TO NODE 3 QUEUE
1510n
NODE X TO NODE 7 QUEUE

Figure 15

1412

A MORE DETAILED LOOK AT THE DEQUEUED_OFFSETS AND THE MESSAGE QUEUES IS SHOWN BELOW:

	0 31	32 63			
0	RESERVED	NODE_OS_ID (EXAMPLES FOLLOW)			1610
		2X	2PN	SCIT	OMUN
1-2	RESERVED	NODE MAC ADDRESS (12 HEX DIGITS WITH 2 DIGITS PER BYTE)			1612
3-7	RESERVED	RESERVED			
0	RESERVED	32 39 RESET_OK	40 47 RESERVED	48 63 DEQUEUED_OFFSET FOR NODE 0	
1	RESERVED	RESET_OK	RESERVED	DEQUEUED_OFFSET FOR NODE 0	
2	RESERVED	RESET_OK	RESERVED	DEQUEUED_OFFSET FOR NODE 0	
			• • •		
7	RESERVED	RESET_OK	RESERVED	DEQUEUED_OFFSET FOR NODE 0	
START OF OUTPUT QUEUE TO NODE 0					P/O NODE-NODE QUEUE 1510a
0	RESERVED	NEED_RESET	RESERVED	ENQUEUED_OFFSET FOR NODE 0	

Figure 16A

1	RESERVED	MESSAGE BUFFER OFFSET		
2	RESERVED	MESSAGE BUFFER OFFSET		
		• • •		
511	RESERVED	MESSAGE BUFFER OFFSET		
START OF OUTPUT QUEUE TO NODE 1				
0	RESERVED	NEED_RESET	RESERVED	ENQUEUED_OFFSET FOR NODE 1
1	RESERVED	MESSAGE BUFFER OFFSET		
2	RESERVED	MESSAGE BUFFER OFFSET		
		• • •		
511	RESERVED	MESSAGE BUFFER OFFSET		
• • •				
START OF OUTPUT QUEUE TO NODE 7				
0	RESERVED	NEED_RESET	RESERVED	ENQUEUED_OFFSET FOR NODE 7
1	RESERVED	MESSAGE BUFFER OFFSET		
2	RESERVED	MESSAGE BUFFER OFFSET		
		• • •		
511	RESERVED	MESSAGE BUFFER OFFSET		

P/O NODE-TO-NODE QUEUE 1510a

1510

1510n

NODE_OS_ID IS A 4 CHARACTER STRING WITH ONE OF THE FOLLOWING VALUES:

- 'OS22' - OS2200 ARCHITECTURE
- 'MCP' - A-SERIES ARCHITECTURE
- 'UNIX' - INTEL ARCHITECTURE WITH A UNIX OPERATING SYSTEM
- 'NT' - INTEL ARCHITECTURE WITH MICROSOFT WINDOWS NT OPERATING SYSTEM

Figure 16B

1416

0	0 RESERVED 31	32 WORD LENGTH OF BUFFER 63	1710
1	RESERVED	WORD LENGTH OF HEADER	
2	RESERVED	32 BYTE SKIP COUNT 47 / 48 BYTE TRANSFER COUNT 63	
3	RESERVED	BYTE SKIP COUNT	BYTE TRANSFER COUNT
n	RESERVED	BYTE SKIP COUNT	BYTE TRANSFER COUNT
m	RESERVED	MESSAGE	1712
	RESERVED		
	RESERVED		
	RESERVED		
	RESERVED		
	RESERVED		
b-1	RESERVED	MESSAGE	

Figure 17

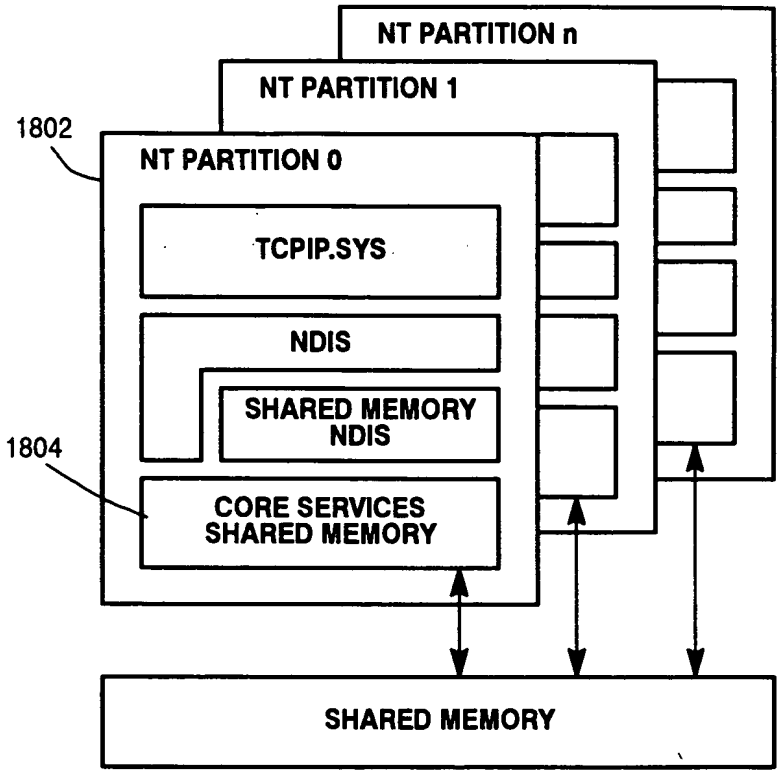
NT PARTITION n
NT PARTITION 1
1802
NT PARTITION 0
TCPIP.SYS
NDIS
SHARED MEMORY
NDIS
1804
CORE SERVICES
SHARED MEMORY
SHARED MEMORY

Figure 18

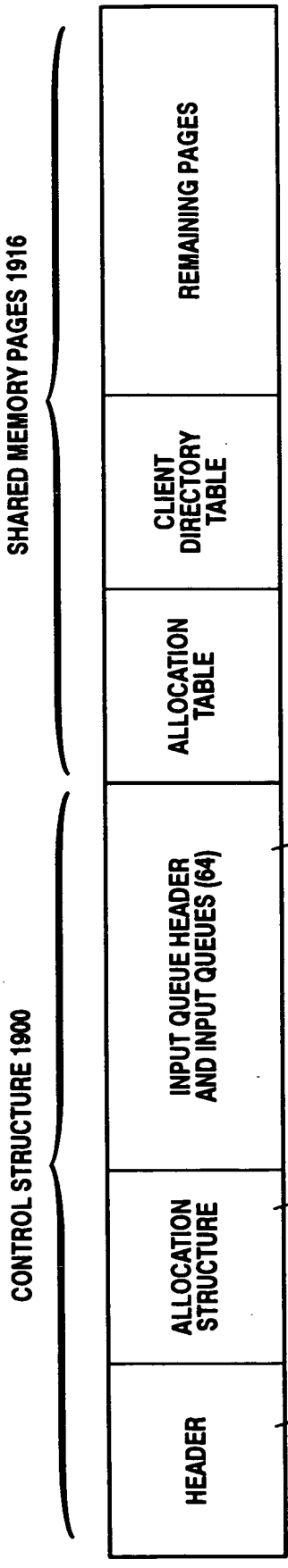
CONTROL STRUCTURE 1900
SHARED MEMORY PAGES 1916
HEADER
ALLOCATION STRUCTURE
INPUT QUEUE HEADER AND INPUT QUEUES (64)
ALLOCATION TABLE
CLIENT DIRECTORY TABLE
REMAINING PAGES
1910
1912
1914

Figure 19

1910

CONTROL STRUCTURE HEADER CONTENTS
VERSION ID
SHARED MEMORY STATUS
PARTITION ID OF "MASTER PARTITION"
SHARED MEMORY PARTITION CHECK IN INTERVAL
CLIENT DIRECTORY TABLE HEADER
PARTITION INFORMATION (10 WORDS PER PARTITION)

Figure 20

1912

ALLOCATION STRUCTURE CONTENTS
ALLOCATION LOCK
LENGTH OF SHARED MEMORY AREA (IN 4K BYTES PAGES)
SHARED MEMORY PAGE POINTER
FREE PAGE LIST HEAD
ALLOCATION TABLE HEADER

Figure 21

USYS-0094 "System and Method for Emulating Network Communications between Partitions of a Computer System"
S.B.Samuels-WoodcockWashburn et al.
215-568-3100

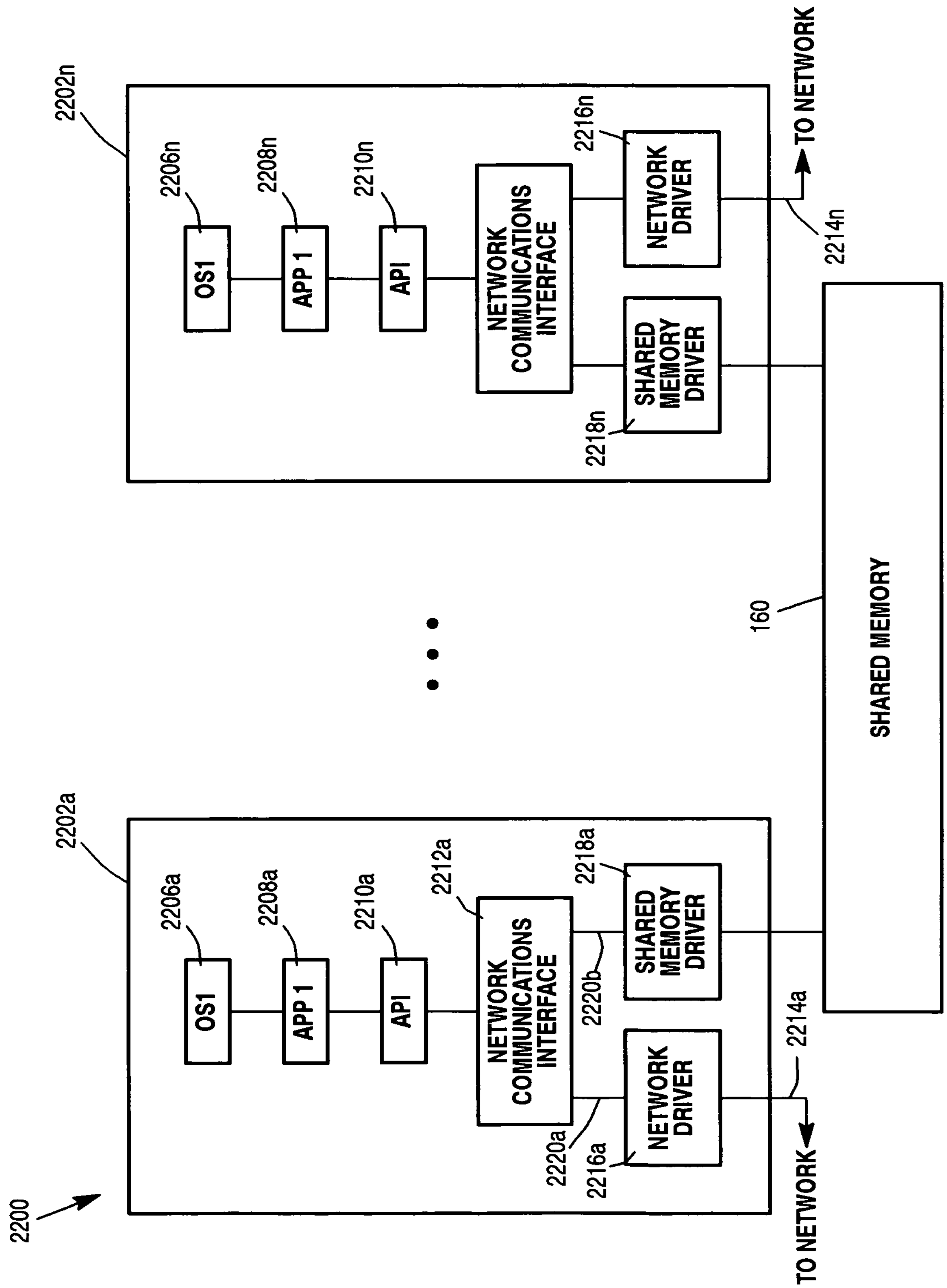

Figure 22

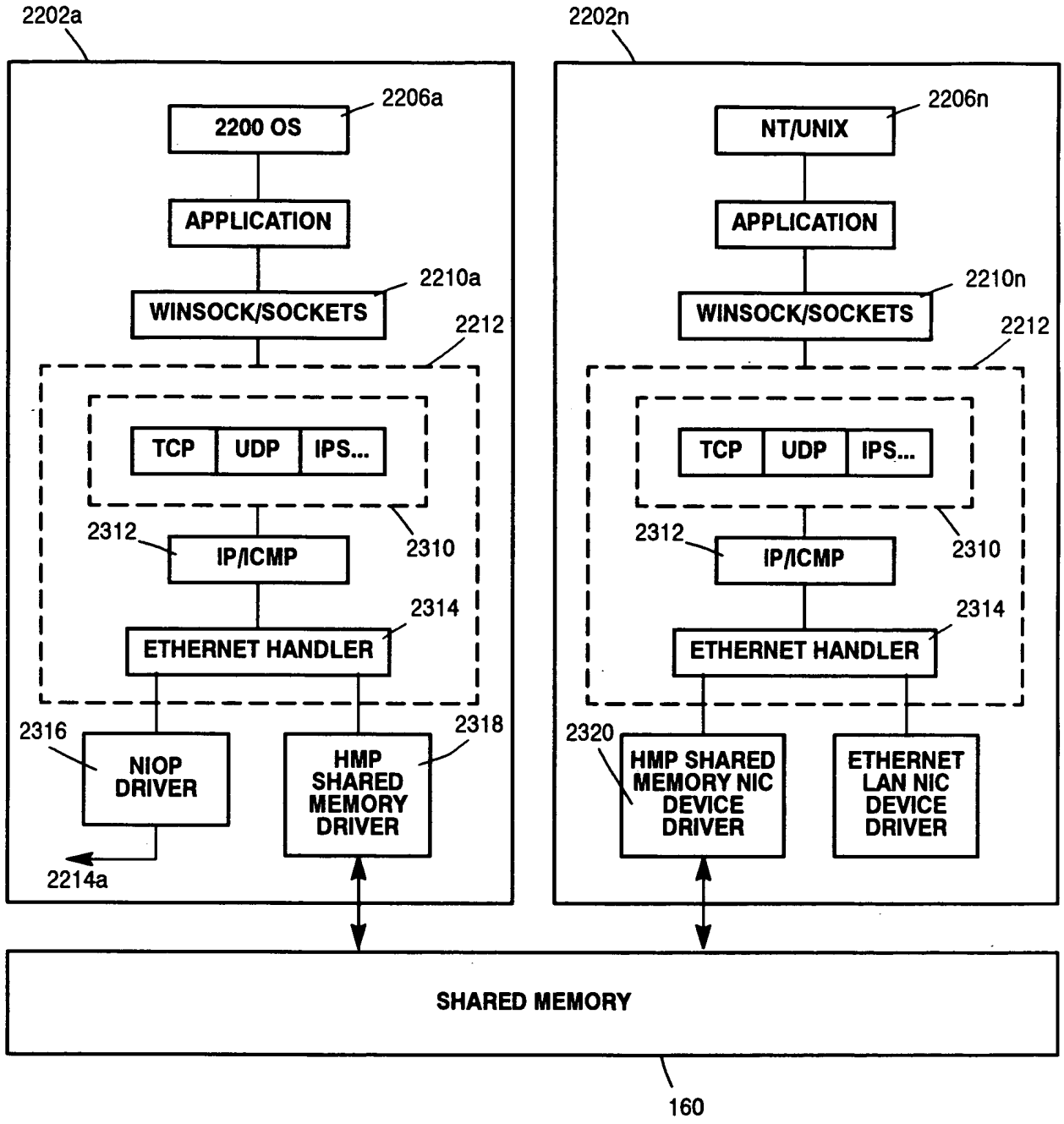
2202a
2202n
2206a
2200 OS
APPLICATION
2210a
WINSOCK/SOCKETS
2212
TCP
UDP
IPS...
2310
2312
IP/ICMP
2314
ETHERNET HANDLER
2316
NIOP DRIVER
2318
HMP SHARED MEMORY DRIVER
2214a
2206n
NT/UNIX
APPLICATION
2210n
WINSOCK/SOCKETS
2212
TCP
UDP
IPS...
2310
2312
IP/ICMP
2314
ETHERNET HANDLER
2320
HMP SHARED MEMORY NIC DEVICE DRIVER
ETHERNET LAN NIC DEVICE DRIVER
SHARED MEMORY
160

Figure 23

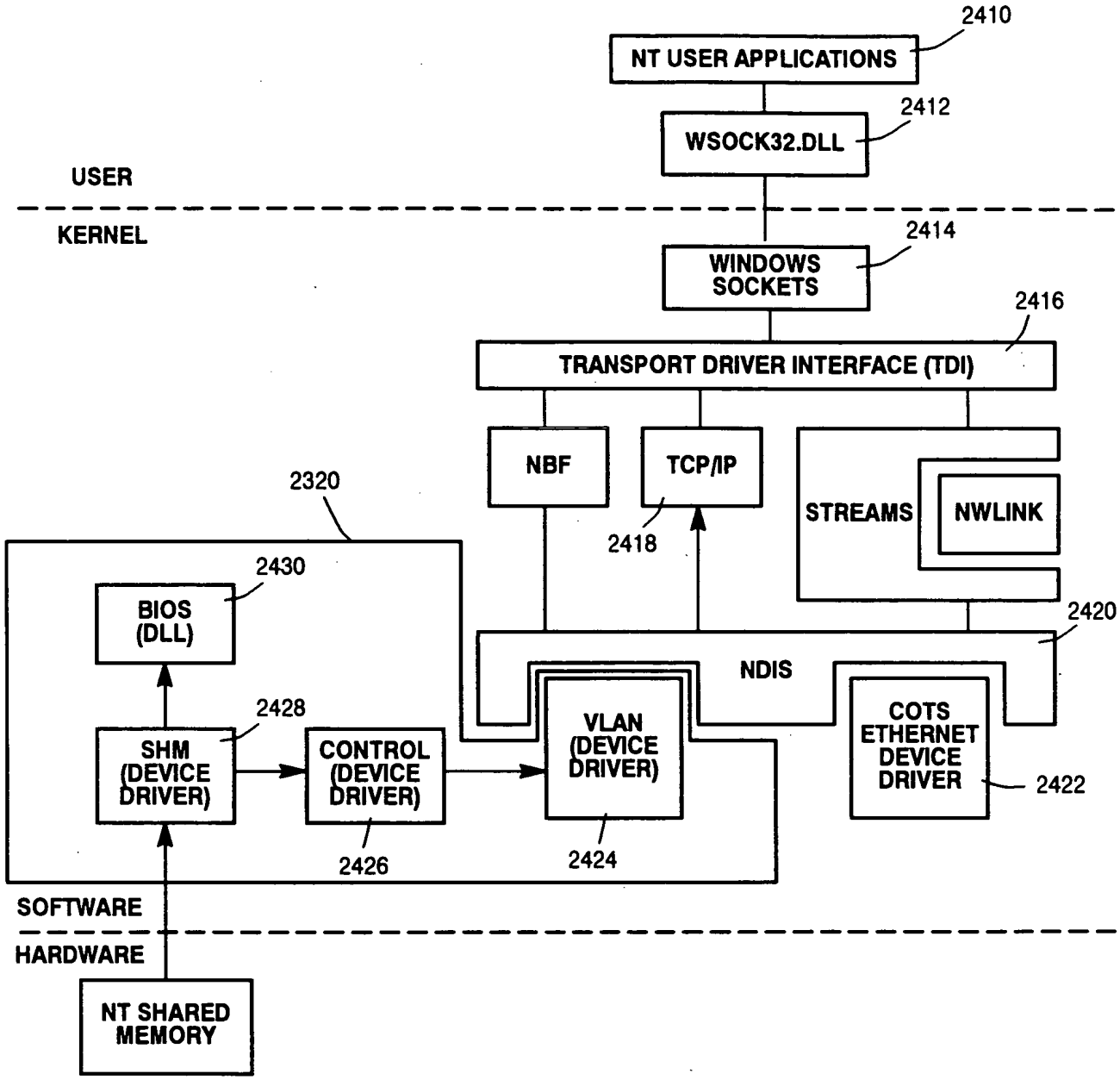
2410
NT USER APPLICATIONS
2412
WSOCK32.DLL
USER
KERNEL
2414
WINDOWS SOCKETS
2416
TRANSPORT DRIVER INTERFACE (TDI)
NBF
TCP/IP
STREAMS
NWLINK
2320
2418
2430
BIOS (DLL)
2420
NDIS
2428
SHM (DEVICE DRIVER)
CONTROL (DEVICE DRIVER)
VLAN (DEVICE DRIVER)
COTS ETHERNET DEVICE DRIVER
2422
2426
2424
SOFTWARE
HARDWARE
NT SHARED MEMORY

Figure 24

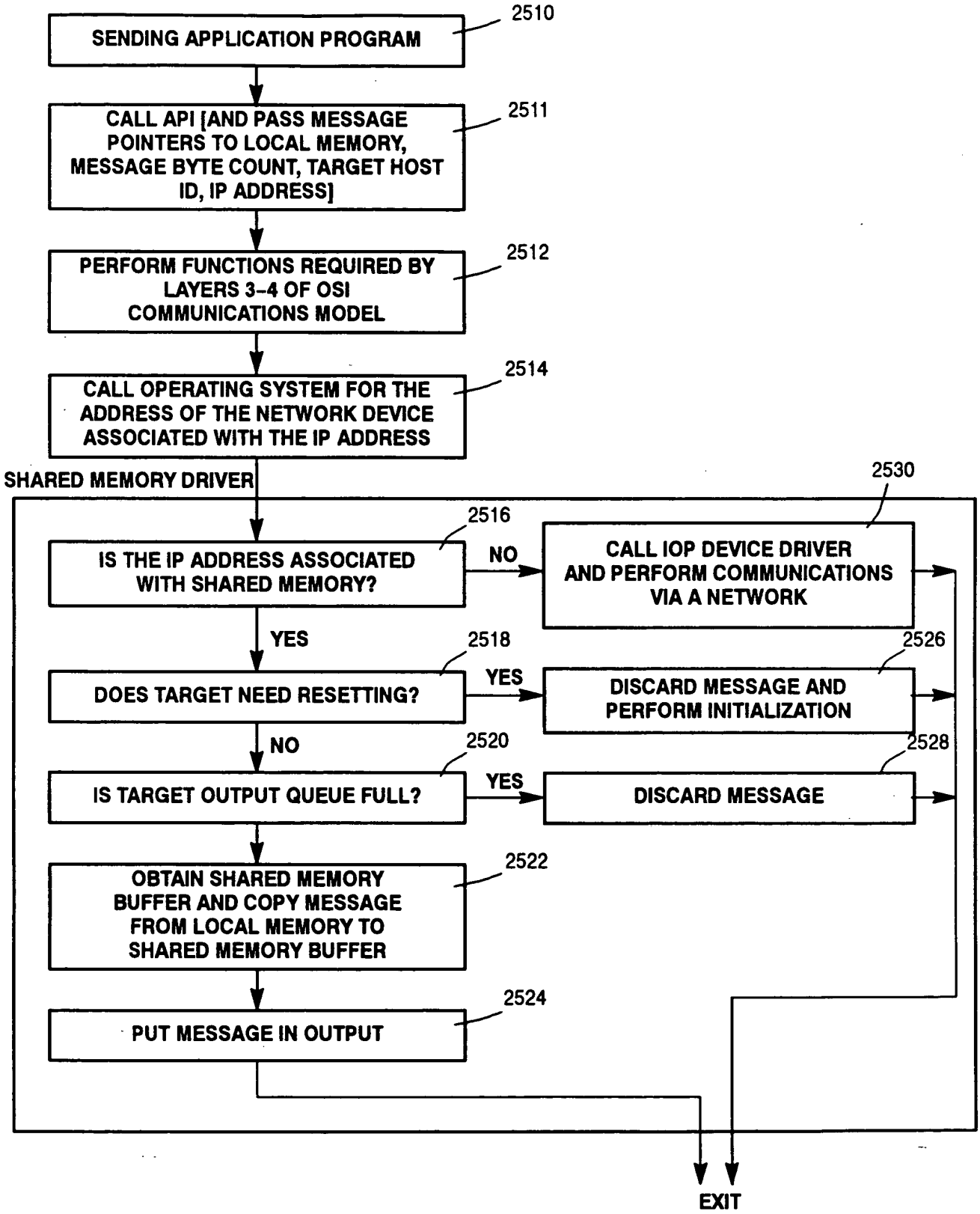
SENDING APPLICATION PROGRAM
2510
CALL API [AND PASS MESSAGE POINTERS TO LOCAL MEMORY, MESSAGE BYTE COUNT, TARGET HOST ID, IP ADDRESS]
2511
PERFORM FUNCTIONS REQUIRED BY LAYERS 3-4 OF OSI COMMUNICATIONS MODEL
2512
CALL OPERATING SYSTEM FOR THE ADDRESS OF THE NETWORK DEVICE ASSOCIATED WITH THE IP ADDRESS
2514
SHARED MEMORY DRIVER
2530
IS THE IP ADDRESS ASSOCIATED WITH SHARED MEMORY?
2516
NO
CALL IOP DEVICE DRIVER AND PERFORM COMMUNICATIONS VIA A NETWORK
YES
DOES TARGET NEED RESETTING?
2518
YES
DISCARD MESSAGE AND PERFORM INITIALIZATION
2526
NO
IS TARGET OUTPUT QUEUE FULL?
2520
YES
DISCARD MESSAGE
2528
OBTAIN SHARED MEMORY BUFFER AND COPY MESSAGE FROM LOCAL MEMORY TO SHARED MEMORY BUFFER
2522
PUT MESSAGE IN OUTPUT
2524
EXIT

Figure 25

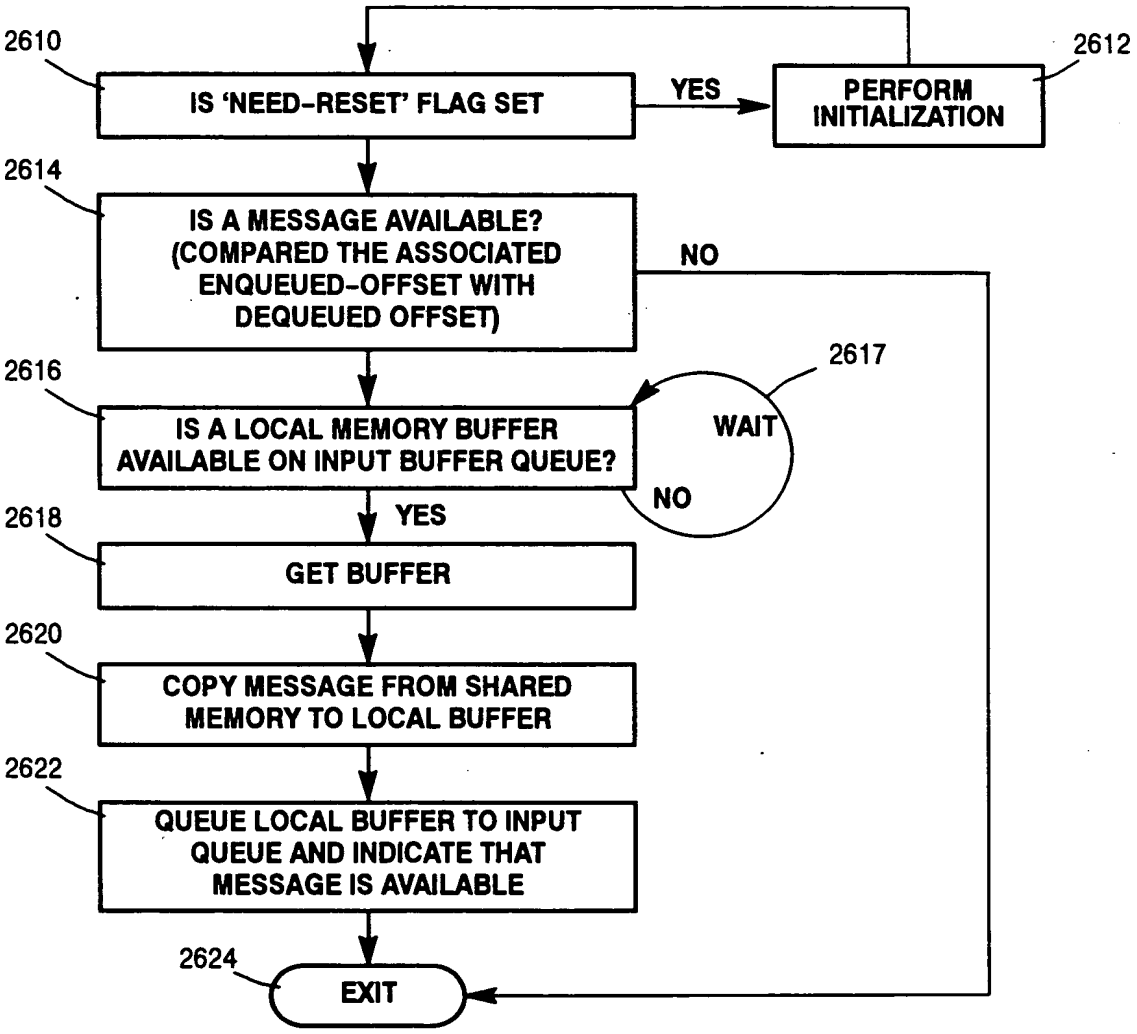
2610
IS 'NEED-RESET' FLAG SET
YES
2612
PERFORM INITIALIZATION
2614
IS A MESSAGE AVAILABLE? (COMPARED THE ASSOCIATED ENQUEUED-OFFSET WITH DEQUEUED OFFSET)
NO
2616
IS A LOCAL MEMORY BUFFER AVAILABLE ON INPUT BUFFER QUEUE?
2617
WAIT
NO
YES
2618
GET BUFFER
2620
COPY MESSAGE FROM SHARED MEMORY TO LOCAL BUFFER
2622
QUEUE LOCAL BUFFER TO INPUT QUEUE AND INDICATE THAT MESSAGE IS AVAILABLE
2624
EXIT

Figure 26

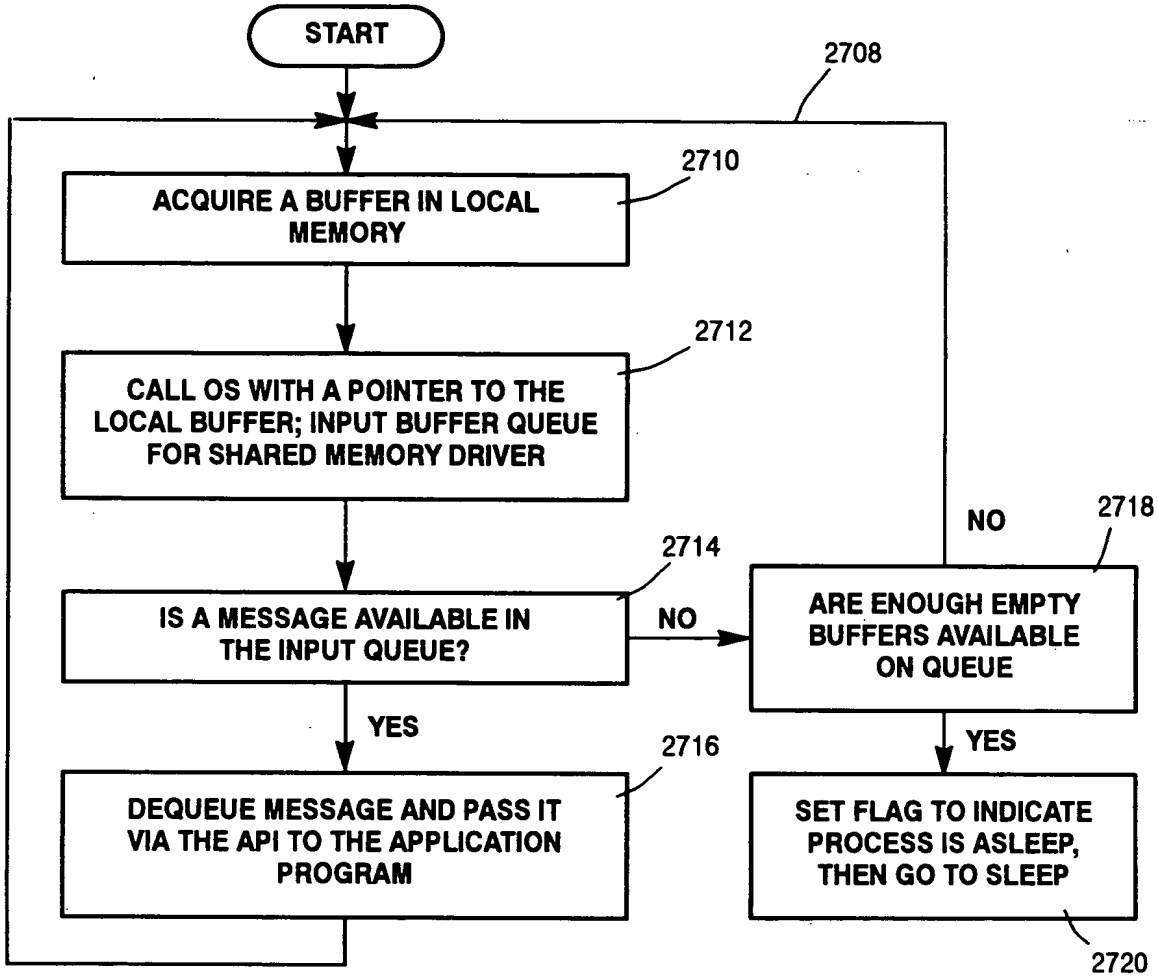
START
2708
2710
ACQUIRE A BUFFER IN LOCAL MEMORY
2712
CALL OS WITH A POINTER TO THE LOCAL BUFFER; INPUT BUFFER QUEUE FOR SHARED MEMORY DRIVER
2714
IS A MESSAGE AVAILABLE IN THE INPUT QUEUE?
NO
2718
ARE ENOUGH EMPTY BUFFERS AVAILABLE ON QUEUE
NO
YES
2716
DEQUEUE MESSAGE AND PASS IT VIA THE API TO THE APPLICATION PROGRAM
YES
SET FLAG TO INDICATE PROCESS IS ASLEEP, THEN GO TO SLEEP
2720

Figure 27

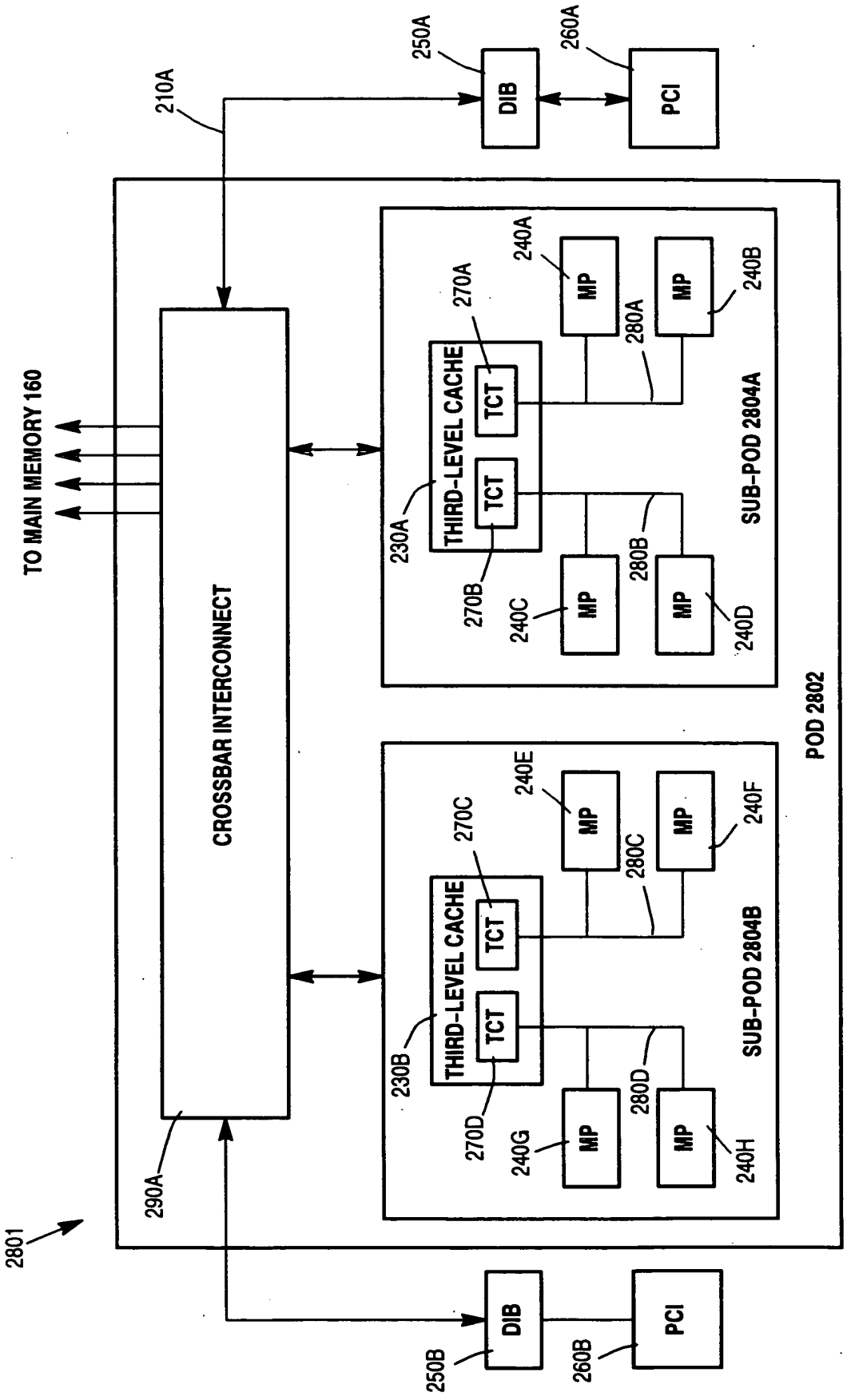
2801
TO MAIN MEMORY 160
CROSSBAR INTERCONNECT
210A
290A
250A
DIB
260A
PCI
250B
DIB
260B
PCI
POD 2802
SUB-POD 2804A
SUB-POD 2804B
THIRD-LEVEL CACHE
TCT
MP
230A
230B
240A
240B
240C
240D
240E
240F
240G
240H
270A
270B
270C
270D
280A
280B
280C
280D

Figure 28

INPUT QUEUE HEADER CONTENTS
INPUT QUEUES POINTER
NUMBER OF INPUT QUEUES
INPUT QUEUE LENGTH
INPUT QUEUE SIGNAL SIZE
MAX NUMBER OF SIGNALS IN INPUT QUEUE

Figure 29

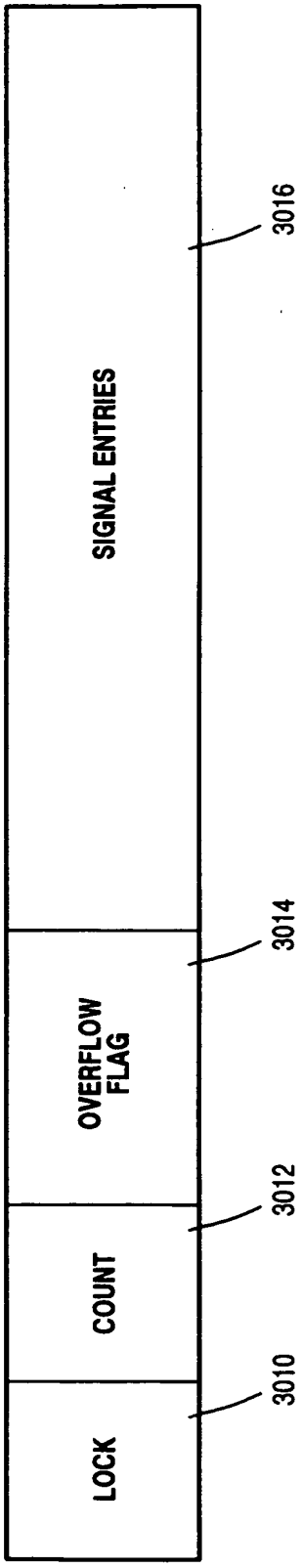
SIGNAL ENTRIES
3016
OVERFLOW FLAG
3014
COUNT
3012
LOCK
3010

Figure 30

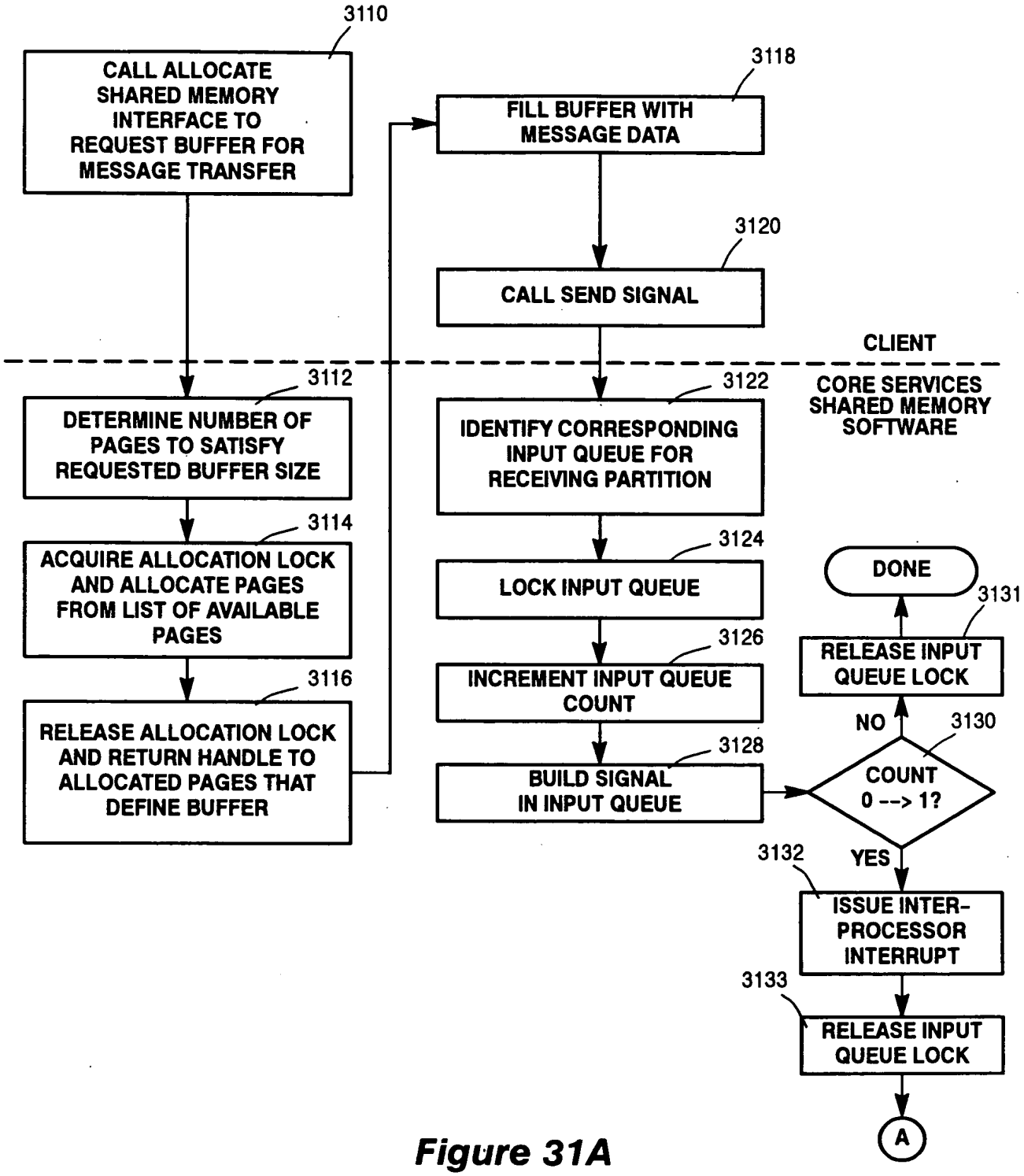
3110
CALL ALLOCATE SHARED MEMORY INTERFACE TO REQUEST BUFFER FOR MESSAGE TRANSFER
3112
DETERMINE NUMBER OF PAGES TO SATISFY REQUESTED BUFFER SIZE
3114
ACQUIRE ALLOCATION LOCK AND ALLOCATE PAGES FROM LIST OF AVAILABLE PAGES
3116
RELEASE ALLOCATION LOCK AND RETURN HANDLE TO ALLOCATED PAGES THAT DEFINE BUFFER
3118
FILL BUFFER WITH MESSAGE DATA
3120
CALL SEND SIGNAL
CLIENT
CORE SERVICES SHARED MEMORY SOFTWARE
3122
IDENTIFY CORRESPONDING INPUT QUEUE FOR RECEIVING PARTITION
3124
LOCK INPUT QUEUE
3126
INCREMENT INPUT QUEUE COUNT
3128
BUILD SIGNAL IN INPUT QUEUE
3130
COUNT 0 --> 1?
NO
YES
3131
RELEASE INPUT QUEUE LOCK
DONE
3132
ISSUE INTER-PROCESSOR INTERRUPT
3133
RELEASE INPUT QUEUE LOCK
A

Figure 31A

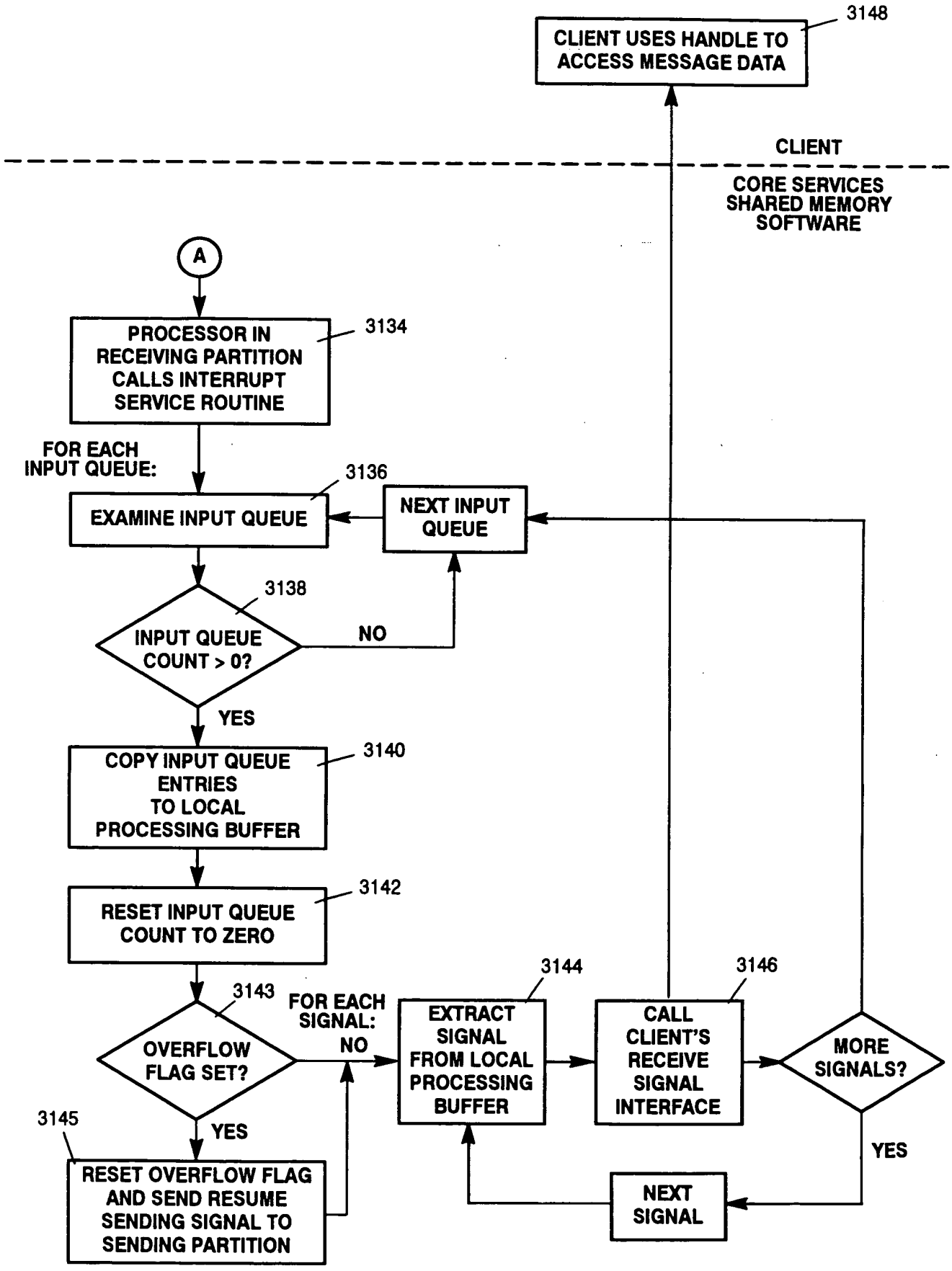
CLIENT USES HANDLE TO ACCESS MESSAGE DATA
3148
CLIENT
CORE SERVICES SHARED MEMORY SOFTWARE
A
PROCESSOR IN RECEIVING PARTITION CALLS INTERRUPT SERVICE ROUTINE
3134
FOR EACH INPUT QUEUE:
EXAMINE INPUT QUEUE
3136
NEXT INPUT QUEUE
INPUT QUEUE COUNT > 0?
3138
NO
YES
COPY INPUT QUEUE ENTRIES TO LOCAL PROCESSING BUFFER
3140
RESET INPUT QUEUE COUNT TO ZERO
3142
OVERFLOW FLAG SET?
3143
FOR EACH SIGNAL:
NO
YES
3145
RESET OVERFLOW FLAG AND SEND RESUME SENDING SIGNAL TO SENDING PARTITION
EXTRACT SIGNAL FROM LOCAL PROCESSING BUFFER
3144
CALL CLIENT'S RECEIVE SIGNAL INTERFACE
3146
MORE SIGNALS?
YES
NEXT SIGNAL

Figure 31B

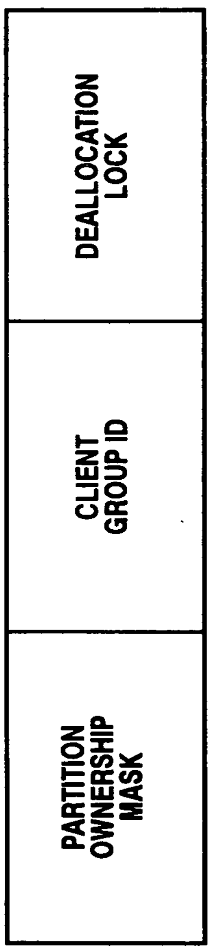

Figure 32A

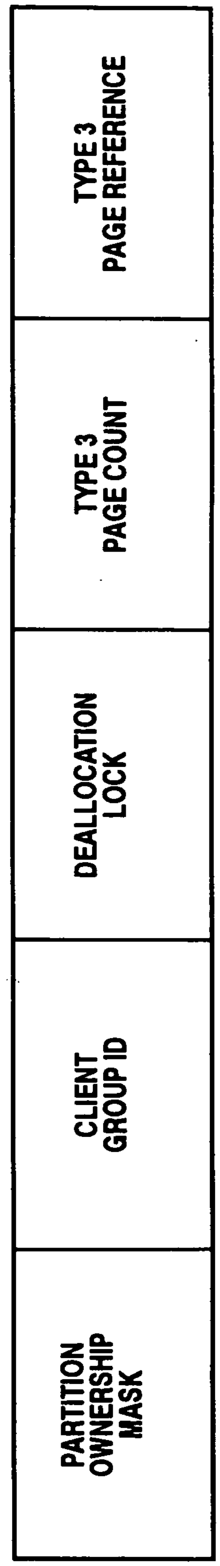

Figure 32B

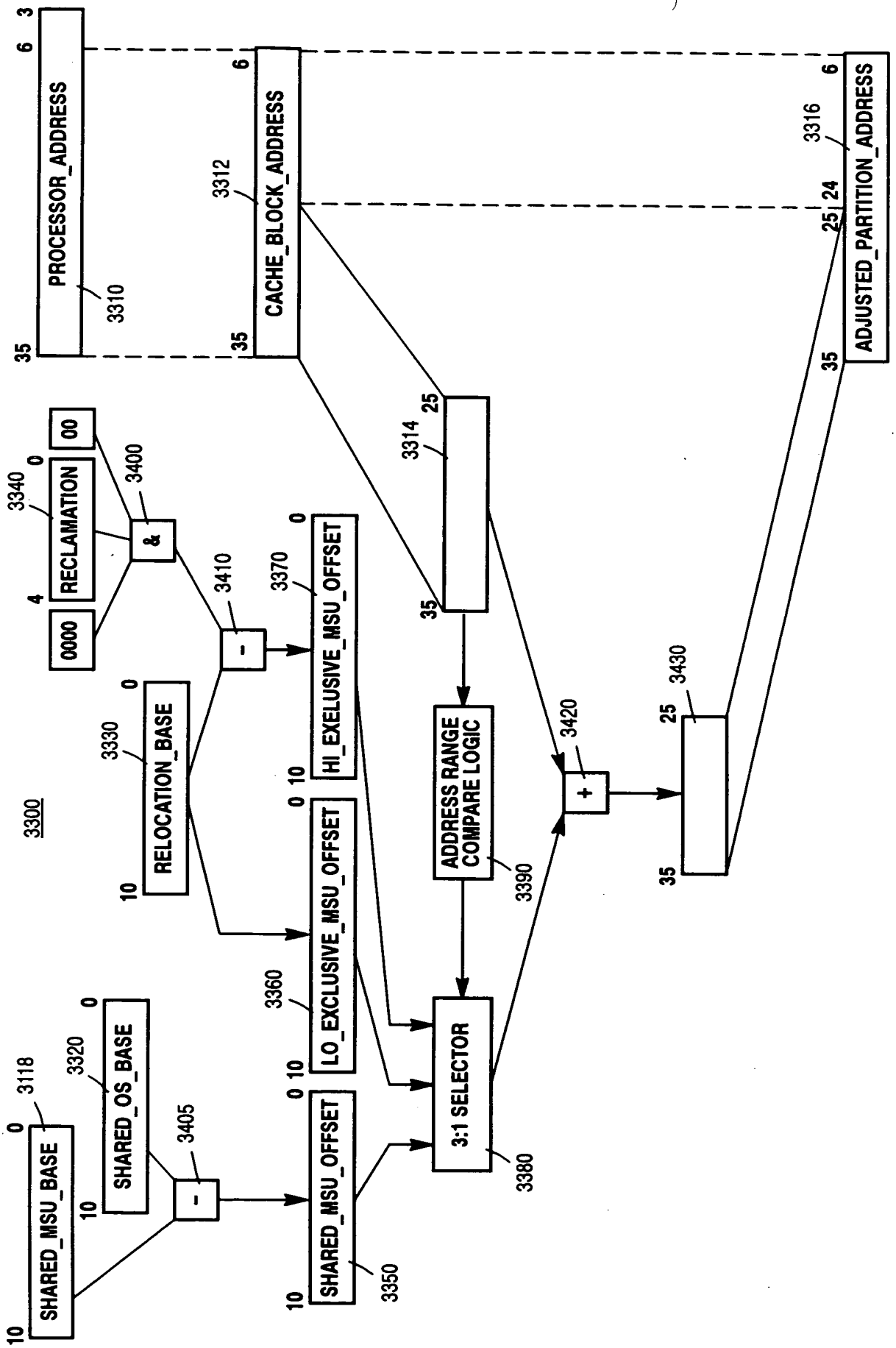
PROCESSOR_ADDRESS
3310
CACHE_BLOCK_ADDRESS
3312
ADJUSTED_PARTITION_ADDRESS
3316
RECLAMATION
3340
00
0000
3400
&
3410
-
RELOCATION_BASE
3330
HI_EXELUSIVE_MSU_OFFSET
3370
LO_EXCLUSIVE_MSU_OFFSET
3360
SHARED_MSU_BASE
3118
SHARED_OS_BASE
3320
3405
SHARED_MSU_OFFSET
3350
3314
ADDRESS RANGE
COMPARE LOGIC
3390
3:1 SELECTOR
3380
3420
+
3430
3300

Figure 33

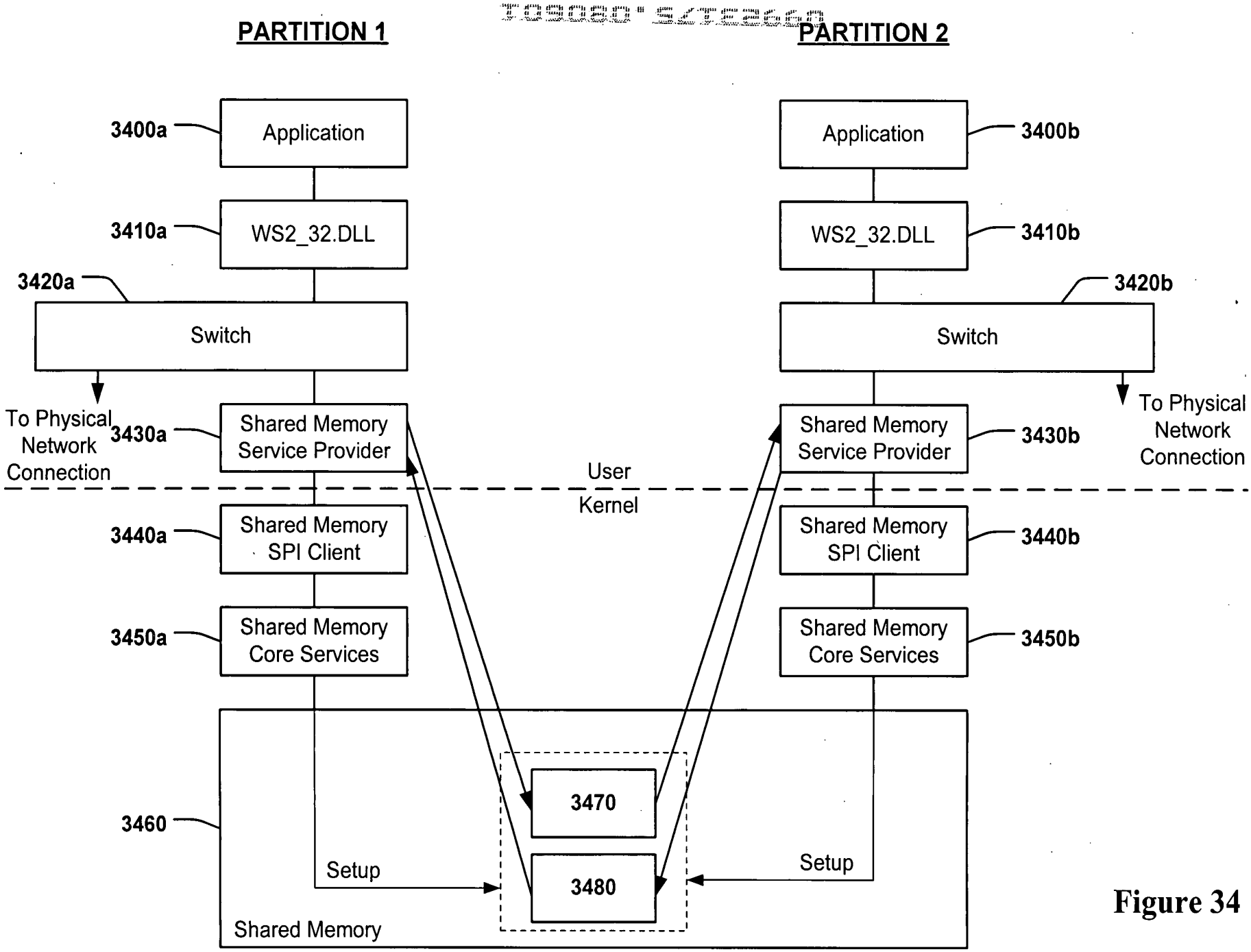

Figure 34

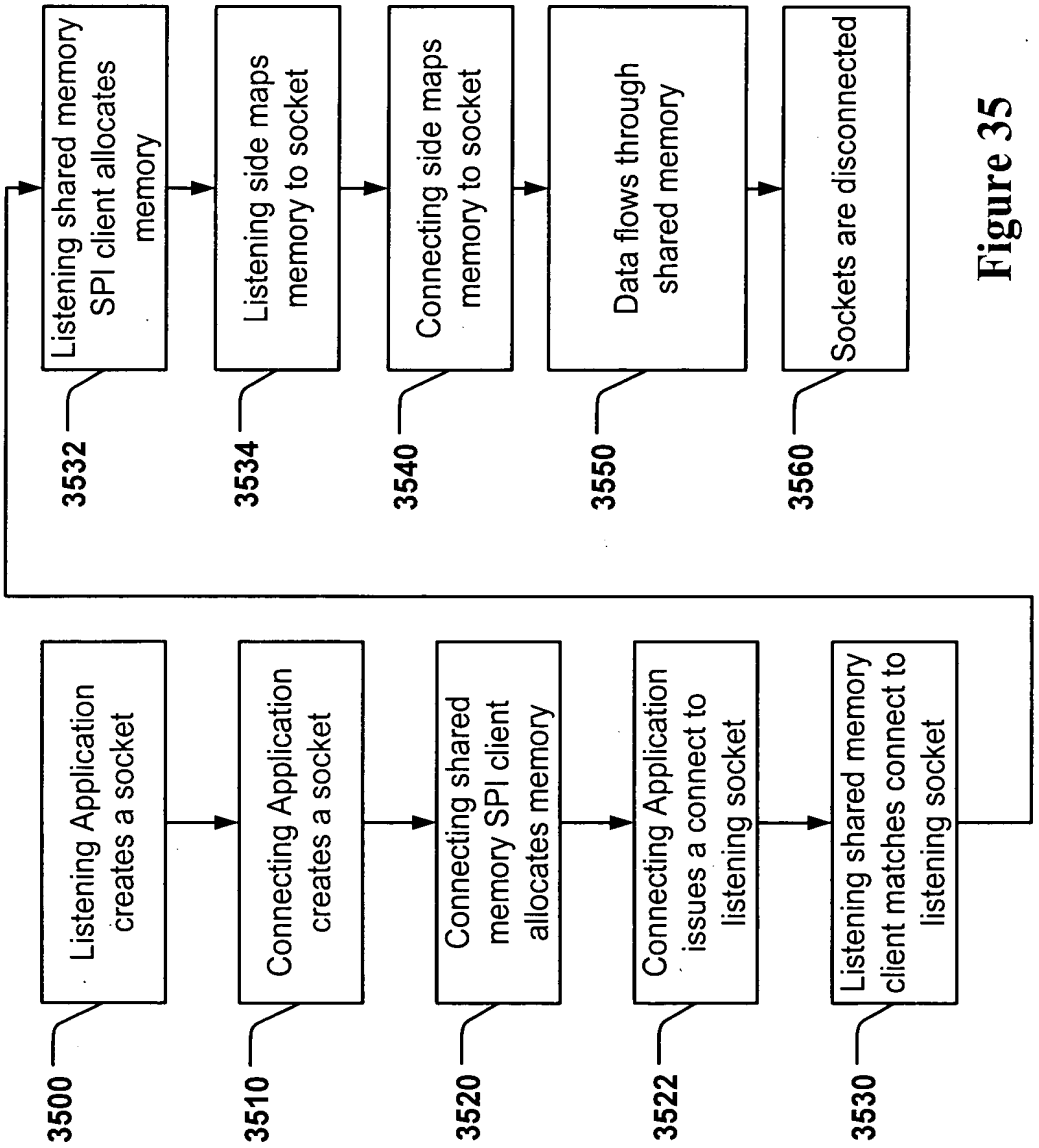
Listening Application creates a socket
3500
Connecting Application creates a socket
3510
Connecting shared memory SPI client allocates memory
3520
Connecting Application issues a connect to listening socket
3522
Listening shared memory client matches connect to listening socket
3530
Listening shared memory SPI client allocates memory
3532
Listening side maps memory to socket
3534
Connecting side maps memory to socket
3540
Data flows through shared memory
3550
Sockets are disconnected
3560

Figure 35

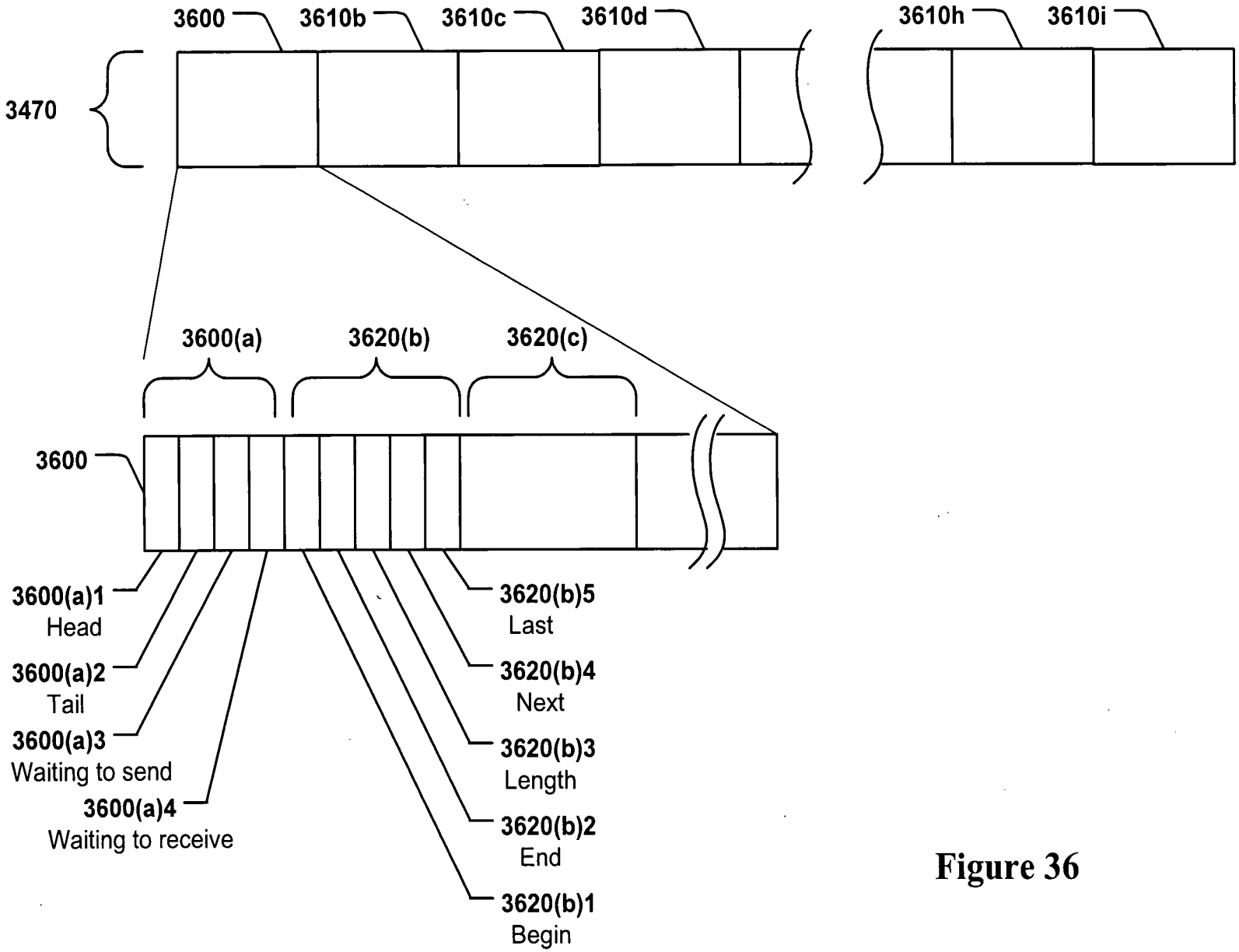
3600
3610b
3610c
3610d
3610h
3610i
3470
3600(a)
3620(b)
3620(c)
3600
3600(a)1
Head
3600(a)2
Tail
3600(a)3
Waiting to send
3600(a)4
Waiting to receive
3620(b)5
Last
3620(b)4
Next
3620(b)3
Length
3620(b)2
End
3620(b)1
Begin

Figure 36

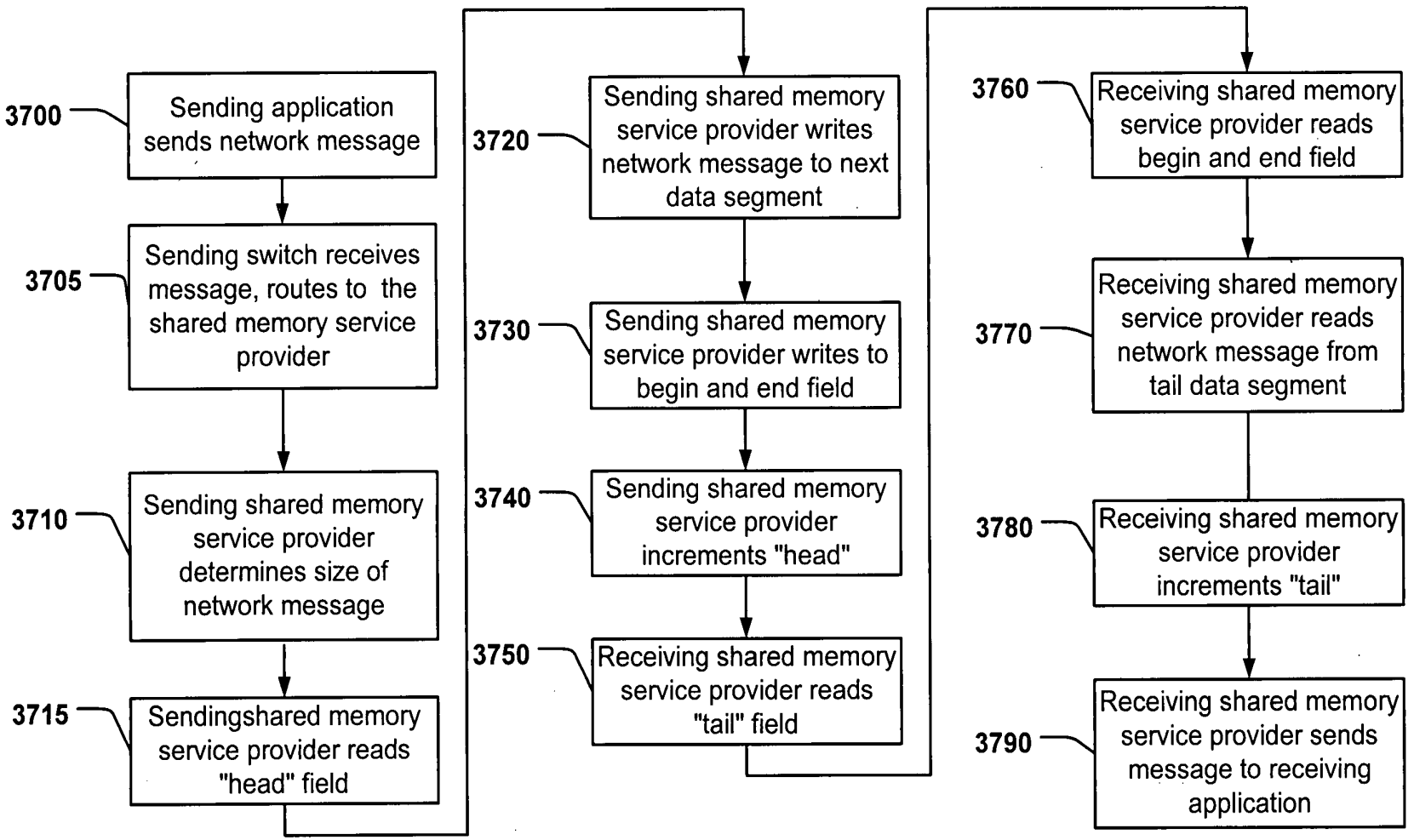
3700
Sending application sends network message
3705
Sending switch receives message, routes to the shared memory service provider
3710
Sending shared memory service provider determines size of network message
3715
Sendingshared memory service provider reads "head" field
3720
Sending shared memory service provider writes network message to next data segment
3730
Sending shared memory service provider writes to begin and end field
3740
Sending shared memory service provider increments "head"
3750
Receiving shared memory service provider reads "tail" field
3760
Receiving shared memory service provider reads begin and end field
3770
Receiving shared memory service provider reads network message from tail data segment
3780
Receiving shared memory service provider increments "tail"
3790
Receiving shared memory service provider sends message to receiving application

Figure 37